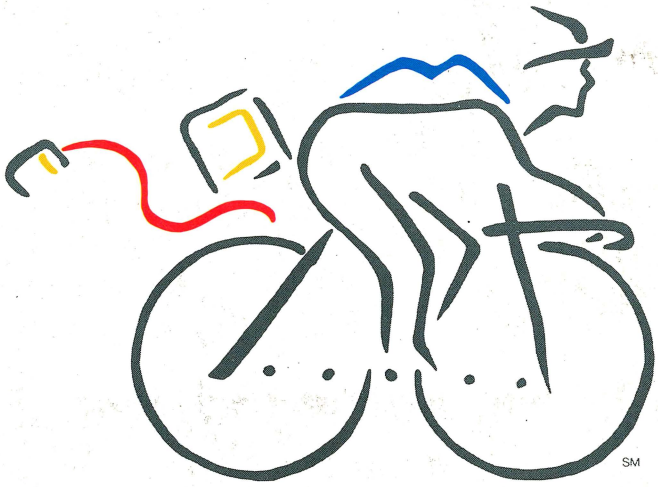
WINTER 1987

Vol.3 No.1
Four Dollars

WHEELS FOR THE MIND

An
Apple University
Publication
Prepared at
Boston College



Detach here and mail.

Save 25% with this introductory offer.

☐ YES! I want to save 25% on a one-year subscription to *Wheels for the Mind*. I understand that if I sign up now, I'll receive four exciting, information-packed issues for just $12. To help us keep you abreast of current developments in your field, please complete all of the following:

Name _____

Address _____

City _____ State _____ Zip _____

Phone No.

Do you own a computer? ☐ Yes ☐ No

If so, what kind? _____

What is your major field of study? _____

Special areas of interest? _____

What sections of *"Wheels"* do you find most interesting/useful?

☐ Editorial
☐ Work in Progress
☐ Macintosh Computer Products
☐ Dialog Box
☐ Technical/Development

X _____
Signature Date

Send no money now. You will be billed later.

Detach here and mail.

Save 25% with this introductory offer.

☐ YES! I want to save 25% on a one-year subscription to *Wheels for the Mind*. I understand that if I sign up now, I'll receive four exciting, information-packed issues for just $12. To help us keep you abreast of current developments in your field, please complete all of the following:

Name _____

Address _____

City _____ State _____ Zip _____

Phone No.

Do you own a computer? ☐ Yes ☐ No

If so, what kind? _____

What is your major field of study? _____

Special areas of interest? _____

What sections of *"Wheels"* do you find most interesting/useful?

☐ Editorial
☐ Work in Progress
☐ Macintosh Computer Products
☐ Dialog Box
☐ Technical/Development

X _____
Signature Date

Send no money now. You will be billed later.

Detach here and mail.

Save 25% with this introductory offer.

☐ YES! I want to save 25% on a one-year subscription to *Wheels for the Mind*. I understand that if I sign up now, I'll receive four exciting, information-packed issues for just $12. To help us keep you abreast of current developments in your field, please complete all of the following:

Name _____

Address _____

City _____ State _____ Zip _____

Phone No.

Do you own a computer? ☐ Yes ☐ No

If so, what kind? _____

What is your major field of study? _____

Special areas of interest? _____

What sections of *"Wheels"* do you find most interesting/useful?

☐ Editorial
☐ Work in Progress
☐ Macintosh Computer Products
☐ Dialog Box
☐ Technical/Development

X _____
Signature Date

Send no money now. You will be billed later.

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 453 CUPERTINO, CA

APPLE COMPUTER, INC.
WHEELS FOR THE MIND 23L
P.O. BOX 810
CUPERTINO, CA 95015

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 453 CUPERTINO, CA

APPLE COMPUTER, INC.
WHEELS FOR THE MIND 23L
P.O. BOX 810
CUPERTINO, CA 95015

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

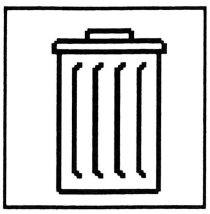
BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 453 CUPERTINO, CA

APPLE COMPUTER, INC.
WHEELS FOR THE MIND 23L
P.O. BOX 810
CUPERTINO, CA 95015

WHEELS FOR THE MIND

Editorial



Hello again and welcome to the Winter issue of ***Wheels for the Mind.***

This issue will focus on the use of the Macintosh computer in the computer sciences. When the "machine for the rest of us" was first introduced, it was felt by many that it would be primarily used in the liberal arts. All of that has certainly changed. With the introduction of the Macintosh Plus computer, the impact of this machine on the sciences became truly significant.

Several of the schools that have adopted the Macintosh computer have integrated it into their computer science curriculum. For many, it is the machine of choice in teaching introductory programming, BASIC, Pascal, assembly language, software development, and systems analysis. Some of these schools even require their students to own a personal computer. In any case, it is clear that the Macintosh computer is not merely the "machine for the rest of us." It may, in fact, one day become the "machine for all of us."

In future issues, we intend to take a look at the use of the Macintosh computer in engineering, business, and education. If you are developing or using any applications in these areas, or would like to submit an article to us, please feel free to do so. Each issue will have the following sections: **Focus**, **Spotlight**, **News and Notes**, **Products**, **Resources**, and **Project Reports**.

We would also like to request more submissions to our **Work Not Yet in Progress** section. It is becoming very popular and can be a great vehicle for sharing your knowledge and expertise with others. Please send your ideas and interests to us.

We prefer receiving material on disk but will accept it in almost any form. Send your material to:

Peter Olivieri
Wheels for the Mind
Fulton 430
Boston College
Chestnut Hill, MA 02167

Take care until next time.

Special Notice About the Wheels for the Mind Awards

The entry form in the fall issue of *Wheels for the Mind* stated that the deadline for contest entries was November 15, 1986. That is **incorrect.**

The **correct** deadline for entries is February 28, 1987, so there's still time—if you act promptly. Please send in your entries to:

Apple Computer, Inc.
10201 De Anza Blvd., M/S 23E
Cupertino, CA 95014
ATTN: Beth Ann Moore

We are sorry for any inconvenience this may have caused.

Wheels for the Mind

Editor:	Peter Olivieri
Copy Editor:	Anne Kaplan-Neher
Contributors:	Beth Ann Moore Kevin Olivieri
Production:	Ron Fernandez Janet Morgan
Coordinators:	Mary Pat Davey Heather Furmidge

Wheels for the Mind is prepared on a Macintosh computer using Aldus PageMaker, and printed out on an Apple LaserWriter printer.

An Apple University publication prepared at Boston College.
Published quarterly.

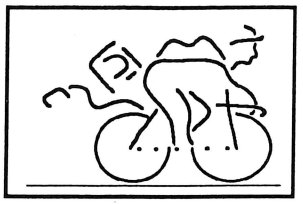
© 1987 Apple Computer, Inc.

Apple, the Apple logo, AppleTalk, ImageWriter, and LaserWriter are registered trademarks of Apple Computer, Inc. MacApp, MacDraw, Macintosh, MacPaint, MacProject, MacTerminal, MacWorks, MacWrite, and Switcher are trademarks of Apple Computer, Inc. The Wheels for the Mind logo is a service mark of Apple Computer, Inc. PageMaker is a registered trademark of Aldus Corporation. PostScript is a registered trademark of Adobe Systems, Inc.

Mention in *Wheels for the Mind* does not constitute an endorsement by Apple Computer. Pricing and availability of third-party products is given according to information received and is subject to change.

Table of Contents

Winter 1987



Focus 1

Focus on Computer Science 1
Computer Science at the Colleges and Universities 2
Macintosh Programmer's Tools Available 3
MacGNOME 5
Waterloo MacJANET for the Apple Macintosh 6
McFace: A Macintosh User Interface for Existing FORTRAN Programs 9
MacScheme and University Education 10
University of Chicago Computer Science Department 14
Computer Science Applications Available from Kinko's 15
Computer Science Applications from Vendors 16

Spotlight 21

University of North Carolina, Chapel Hill 21
MIT: LISP Application 22
University of Utah 23
University of Waterloo 24
Princeton University 27
University of Minnesota 28
Boston College 30
Virginia Tech 31
University of Illinois, Champaign/Urbana 33

News and Notes 41

Apple Programmer's and Developer's Association 41
APDA Membership Agreement and Application 43
MacApp Developer's Association 46
Poor Tom: An Associational Data Base 47
University of Utah Department of Music 52
Top of the Morning to You, Macintosh 54
Macintosh Simulated Camera 55
Desktop Training: Interactive Authoring 57
Edudisc Authoring Work in Progress 65
Work Not Yet in Progress 66
Organic Fonts 1.1 Update 73
Drexel University Publication Development Center 74

Technical Information 77

MacApp: A New Approach to Application Development 77
Macintosh 68000 Development System, Version 2.0 78
Object Pascal 81
Repairing MacWrite Documents 81
Oh No, My File !! 84
Smalltalk-80 for the Macintosh, Release 0.3 88

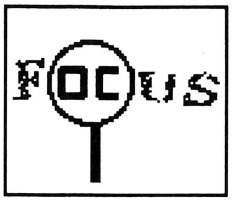
Table of Contents

Macintosh Products **91**
- MacNet Update 91
- MacSpin—The Data Analyst's Wheels for the Mind 93

Resources **103**
- Macintosh Books 103
- Articles of Interest 105
  - Business 105
  - Computer Languages 107
  - Data Bases 109
  - Desk Accessories 110
  - Education 111
  - Games 111
  - Graphic Design 113
  - Hardware 115
  - Miscellaneous 116
  - Music 118
  - Spreadsheets 118
  - Telecommunications 119
  - Utilities 120
  - Word Processing 122
- Additional Macintosh Resources—*MacInTouch* Newsletter 125

Project Reports **129**
- Astronomy 130
- Biology 130
- Chemistry 133
- Computer Science 139
- Design 140
- Economics 141
- Education 141
- Engineering 145
- Geology 152
- History and Politics 153
- Humanities 154
- Languages 156
- Medicine 160
- Miscellaneous 163
- Music 168
- Networks 170
- Nutrition and Food Science 176
- Philosophy 176
- Physics 177
- Psychology 182
- Sociology 183
- Telecommunications 184
- Word Processing 187

Focus



The **Focus** section will concentrate on a different theme in each *Wheels for the Mind* issue. The focus for this issue will be on using the Macintosh in computer science. Among the applications that will be discussed are programming tools, FORTRAN, LISP, Apple University Consortium software, and third-party software. Please send along information on any applications that you feel deserve special attention from those in the academic community.

Focus on Computer Science

By Beth Ann Moore, Apple Computer University Marketing

Computer science. To many of us (especially us liberal arts graduates), this phrase used to evoke images of people sitting around in a room filled from floor to ceiling with giant mainframes, wheels of tapes turning, and punch cards scattered around the floor like they are at the end of the day at the Chicago Commodities Market. (At the end of the semester, cards were thrown by the stacks out of dorm windows!) Language names such as *COBOL, FORTRAN, Pascal, BASIC,* and *UNIX* seemed like secret codes. Words such as *linker, compiler, assembler, pointer, parse table, shell,* and *environment* meant very little. This rather primitive view of computer science was compounded by the fact that few of us actually knew anyone who was involved in this field.

Times have changed—and so has computer science.

But times have changed—and so has computer science. With the advent of sophisticated, powerful, graphically oriented microcomputers and easy-to-learn software, more students are studying computer science than ever before. The Apple® Macintosh™ personal computer and programs such as Macintosh Pascal are being used in many institutions around the country to introduce students to programming, and to light the way for them to begin to create their own programs.

Computer science is an art, just like dance, design, or music.

Suddenly, it becomes obvious that computer science is an *art,* just like dance, design, or music. Skillful—and artistic—use of powerful programming languages produces applications that "the rest of us" can use quickly and efficiently. Software that streamlines common tasks, and software that helps us see things in ways never before possible.

In this issue of *Wheels for the Mind,* you will read about computer science projects that are under way at many different institutions. Gathering this information was extremely interesting, because it meant meeting some of the people behind the projects. I developed a tremendous amount of respect for these folks; their visions and their ability to use the computer to translate these visions into a form that can help us learn, expand our own viewpoints, and experience the world in a new way are almost magical.

Computer Sciences at the Colleges and Universities

Across the country, many institutions have found the Macintosh to be the ideal computer for use in various aspects of the computer science curriculum. Below is a partial listing of the schools using Macintosh in the area of computer science. For a more complete description of some of them, see the **Project Reports** section of this issue of *Wheels for the Mind.*

Baylor University

Boston College

Bowling Green State University

Brigham Young University

California State University at Fresno

Carnegie-Mellon University

Dartmouth College

Drake University

Drexel University

Eastern Washington University

George Washington University

Linfield College

Massachusetts Institute of Technology

Mission College

Northeastern University

Northwestern University

Ohio State University

Oregon State University

Pepperdine University

Princeton University

Reed College

Rice University

Seattle University

Stanford University

Towson State University

Tulane University

University of California at Davis

University of California at Los Angeles

University of California at Santa Barbara

University of Chicago

University of Colorado

University of Illinois

University of Iowa

University of Kansas

University of Maryland

University of Michigan

University of Minnesota

University of North Carolina at Chapel Hill

University of Oregon

University of South Carolina

University of Utah

University of Waterloo

University of Wisconsin

Utah State University

Virginia Polytechnic Institute

Washington University (St. Louis)

Western Oregon State College

West Virginia University

Yale University

Macintosh Programmer's Tools Available

Copies of MPW, MPW Pascal, MPW C, MacApp™, and Smalltalk are now available through the Apple Programmer's and Developer's Association (APDA). (For further information about this organization and how to join, see the **News and Notes** section.) The packages consist of final 1.0 software and drafts of the corresponding manuals. So far, reaction to these products has been extremely positive.

The Macintosh Programmer's Workshop products, hardware and software requirements, and APDA prices are briefly discussed below:

Macintosh Programmer's Workshop (MPW)

This is a complete professional-level, integrated assembly-language programming environment for the Macintosh. This product includes a combined editor and command interpreter, assembler, linker, debugger, Macintosh ROM interfaces, resource editor,

resource compiler/decompiler, full documentation, and a wide variety of utility programs. Translation utilities are provided to convert MDS and Lisa® assembler source code. This package contains five single-sided disks and roughly 700 pages of documentation. Price: $100.

To use the Macintosh Programmer's Workshop, you'll need a Macintosh, or Lisa running MacWorks™, with at least 512K of RAM and 1.6 megabytes of disk storage. A hard disk and 1 megabyte of RAM are highly recommended.

Macintosh Programmer's Workshop Pascal (MPW Pascal)

This is a complete, professional-level, programming environment.

This package contains a native Macintosh Pascal compiler, run-time library, Macintosh interface libraries, full documentation, and sample programs. Pascal formatter and cross-reference tools are also included. Object modules created by the Pascal compiler are link-level compatible with MPW assembler and C modules. The compiler supports the object extensions required by MacApp. This package contains two single-sided disks and roughly 275 pages of documentation. Price: $75.

To use MPW Pascal, you'll need a Macintosh, or Lisa running MacWorks, with at least 512K of RAM and 1.6 megabytes of disk storage. A hard disk and 1 megabyte of RAM are highly recommended. The Macintosh Programmer's Workshop is also required.

Macintosh Programmer's Workshop C (MPW C)

This package contains a native Macintosh Pascal compiler, Standard C library, Macintosh interface libraries, full documentation, and sample programs. Object modules created by the C compiler are link-level compatible with MPW assembler and Pascal modules. This package contains two single-sided disks and roughly 300 pages of documentation. Price: $75.

To use MPW C, you'll need a Macintosh, or Lisa running MacWorks, with at least 1 megabyte of RAM and 1.6 megabytes of disk storage. A hard disk is highly recommended. The Macintosh Programmer's Workshop is also required.

MacApp

MacApp implements the standard features common to most Macintosh programs.

MacApp, an object-oriented programming library, implements the standard features common to most Macintosh programs. It automatically handles standard features such as scrolling, resizing, and printing. MacApp programs are written using MPW Pascal and/or the MPW assembler. This package contains four single-sided disks and roughly 450 pages of documentation (source-code listings are available separately from APDA). Price: $50.

To use MacApp, you'll need a Macintosh, or Lisa running MacWorks, with at least 1 megabyte of RAM and a hard disk. The Macintosh Programmer's Workshop and MPW Pascal are also required.

MacGNOME

MacGNOME is a project at Carnegie-Mellon University devoted to developing integrated novice programming environments. These environments are tailored to the specific needs of the novice programmer, yet they can be helpful to the experienced programmer as well. MacGNOME environments are targeted for affordable microcomputers, and the first implementation is for the Apple Macintosh personal computer. For best results, MacGNOME requires a Macintosh 512K and an external drive.

MacGNOME is geared to the needs of the novice programmer.

The goals of MacGNOME are as follows:

- To provide an integrated environment so the novice programmer does not have to switch between tools with different command interfaces.
- To facilitate quick and accurate feedback, which is extremely helpful to the novice.
- To ensure syntactic correctness while editing, so the novice can concentrate on semantics.

How It Works

MacGNOME provides an environment in which the editor, compiler, and run-time and debugging systems are all integrated transparently to the user via the uniform interface of a structure editor. The technology behind structure editors allows new and exciting forms of user interaction that not only guide the novice, but also stimulate the exploration of new topics and ideas.

This technology allows new and exciting forms of user interaction.

MacGNOME includes the following features:

- Unites edit, compile, run, and debug functions in one system.
- Facilitates teaching of *abstraction*, a fundamental concept in computer science, by using multiple program views.
- Provides immediate feedback—the editor knows about the language being edited and can provide intelligent help and guidance.
- Utilizes state-of-the-art computer science technology (MacGNOME is a leading research project in software engineering).

Applications

Two MacGNOME environments are currently under development for the Macintosh:

Several useful applications for MacGNOME already exist.

Karel the Robot: Based on a language developed by Rich Pattis at Stanford University, Karel teaches fundamental concepts of programming without the burden of a complex language. The Karel environment will be used in elementary and secondary schools, as well as in the first two weeks of a college introductory computer science course.

Pascal: Pascal is the language used by the College Board to deliver its advanced placement computer science course. Pascal is also used to teach introductory programming at most of the leading universities in the country, including Carnegie-Mellon (CMU).

MacGNOME was used to teach introductory programming at CMU in the summer and autumn semesters of 1986. Karel will be available for external use in the spring of 1987, with Pascal following in summer 1987. The MacGNOME environments are expected to cost less than $30. The development of MacGNOME environments is a joint project of the Center for Art & Technology and the computer science department at CMU.

For further information, contact:

Rob Chandhok or Phil Miller
Carnegie-Mellon University
Pittsburgh, PA 15213
(412) 268-3342

Waterloo MacJANET for the Apple Macintosh

Why a Network?

The use of the Apple Macintosh personal computer in an introductory course on computer literacy at the University of Waterloo is just one example of the educational community's growing acceptance of the Macintosh. But such courses require administering the distribution of large numbers of disks containing the necessary applications and courseware. And more problems arise when the course instructor wishes to update the many student disks during the term.

JANET is networking software designed to meet the needs of educational environments

To alleviate such difficulties and to improve the effectiveness of the Macintosh as a teaching tool, software was developed by the University of Waterloo to connect the stand-alone machines to form a network. Waterloo JANET, originally developed on the IBM PC, is networking software designed to meet the needs of educational environments. The system supports a large number of users through workstations connected to a central file and print server.

Waterloo MacJANET for the Apple Macintosh consists of a Macintosh equipped with a hard disk as the file server and a number of Macintosh workstations connected to the network server. The workstations (which are vanilla Macintosh systems), are connected to the server using Apple's built-in AppleTalk® hardware.

The network-server hard disk is shared among the Macintosh workstations, to allow the sharing of applications and data. The students' private files are also stored on the file server. Files sent to an AppleTalk ImageWriter® and/or LaserWriter® printer may temporarily reside (that is, be spooled) on the hard disk while awaiting print.

The User's Perspective

A student gains access to the system through the Waterloo MacJANET desk accessory, which is located on the Macintosh startup disk. Along with the user name, a password must be specified for protection of private data. This action, which is called "accessing a network disk," automatically provides access at the workstation to the individual's files on the shared Macintosh network server. These files appear on the desktop just as if the user had inserted a regular disk into a disk drive attached to the Macintosh.

A password must be specified for protection of private data.

Network disks are used in the same fashion as real disks located in either the internal Macintosh drive or an external drive. These network disks are represented by a special disk icon that may appear in one of four forms—accessed, selected, opened, or opened and selected—according to its state.

In addition to their personal network disks, students may use the Waterloo MacJANET desk accessory to access other network disks, such as public-library disks containing applications software or courseware. These shared public-library disks are accessed in read-only mode, indicated by a padlock icon displayed in the title bar of the disk directory window. In addition, Macintosh application files can be marked as protected, so that students can execute an application but not copy the program.

The startup disk (system disk) in the Macintosh workstation is the default disk even while executing applications. Temporary work files are created locally on this disk, reducing the size required for a user's private network disk. This arrangement also reduces overall traffic on the network, thereby increasing its capacity.

When a student finishes working with the Macintosh, he or she uses the Waterloo MacJANET desk accessory to instruct the system to log off. This releases any network disks accessed during the student's session, leaving the workstation ready for the next student.

An important design goal of the network was transparency to the user. Normal Macintosh procedures are used to work with the

network disks, which makes the process as familiar as using real disks in the actual internal and external disk drives.

The Administrator's Perspective

An administrator's workstation provides extra security against unauthorized use.

The Waterloo MacJANET administration program is located on the system administrator's network disk. The program is run on a Macintosh workstation while the network is running, thus eliminating the need to shut down the system. This important program contains an added level of security to protect against unauthorized use.

The administration program executes a number of key functions with minimal intervention from the network administrator. These functions include directory maintenance, manual or automatic generation of user accounts, and collection of user information. Information is gathered in the user directory file, including user name, password, the size of network disk allocation, and restrictions on access to other network disks. The network system administrator can easily modify the information contained in this file by adding or removing users, or by changing the characteristics of current users, through the administration program.

An individual's access can be restricted to certain classes of network disks.

An additional convenient feature is that individual user accounts can be configured to let the users automatically access particular network disks whenever they log onto the system. Also, an individual's access may be restricted to certain classes of network disks. This added restriction ensures security of sensitive information on the network disk.

The network administration facilities are designed to cope with procedures peculiar to the educational environment. An example of this is automatic creation or deletion of large numbers of student accounts.

Use at the University of Waterloo

Waterloo MacJANET has been in production use at the University of Waterloo for over a year. The production system is actually made up of three separate networks, each consisting of one Macintosh 512K with a built-in 20-megabyte HyperDrive hard disk from General Computer Company, and 15 Macintosh 512K workstations attached via the AppleTalk network. ImageWriter printers are attached directly to workstations to provide printed output.

The need to handle thousands of disks has been eliminated.

In Waterloo MacJANET's first year at the University of Waterloo (prior to release), more than 1,500 students used the system through the computer literacy course. Waterloo MacJANET appears to have relieved the computer-related administrative problems, eliminating the need to handle thousands of disks for the students. Initial reactions of the course professors, teaching assistants, and students have been very positive. In other areas of the university and in other educational institutions, interest in the network as a solution to the problem of networking stand-alone Macintosh systems to form a shared environment is growing.

Carl Durance, director of the Waterloo MacJANET project, is associate director of the Computer Systems group at the University of Waterloo.

For information on purchasing MacJANET, site licensing, maintenance, and upgrades, contact:

WATCOM Products, Inc.
415 Phillip Street
Waterloo, Ontario
Canada N2L 3X2
(519) 886-3700
Telex: 06-955458

McFace: A Macintosh User Interface for Existing FORTRAN Programs

By Dan Kampmeier
University of Illinois at Urbana-Champaign

The Interface Gap

Absoft did an excellent job of bringing FORTRAN to the Macintosh, and conversion of existing FORTRAN programs for use on the Macintosh is relatively simple and straightforward. Unfortunately, running such programs without further modification is disappointing to the average Macintosh user, who has come to expect a great deal of the Macintosh user interface. But a typical FORTRAN programmer is more concerned with writing code to crunch numbers and/or plot results than with learning the idiosyncrasies of the Macintosh Toolbox. Moreover, of the major programming languages available for the Macintosh, FORTRAN has been given the least support with respect to use of the ROM Toolbox routines. McFace was developed to help bridge this "interface gap," the gap between Macintosh user expectations and existing FORTRAN programs.

Bridging the Gap

This program adds menus and windows to your FORTRAN programs.

McFace is a FORTRAN-callable subroutine that creates a Macintosh user interface for existing FORTRAN programs. It adds, without a single Toolbox call by the calling program, the following features:

- Apple, File, Edit, and Window menus for access to desk accessories, file handling, text editing, picture handling, and window management.
- Up to five scrollable, resizable text-editing windows that can be used as the destination of FORTRAN output or the source of FORTRAN input.

- A resizable window (up to 30,000 x 30,000 pixels) for graphical output in which drawings can be recorded as bit maps (like MacPaint™), pictures (like MacDraw™), or any combination of the two types.

- Simple access to and manipulation of menus, alerts, and dialog boxes created by the calling program.

A Typical Conversion

Conversion of a typical program for use with McFace could involve changing text I/O to work with McFace's text editors, creating new menus and alert boxes to substitute for simple interactive text I/O, bringing data from input files into dialog boxes (which McFace lets you do directly), and sending graphical output to McFace's graph window. The resulting program would have easily editable text and numerical input and output; would be controlled by mouse-driven menus, alerts, and dialog boxes; and would produce graphical output that can be easily moved to other drawing programs for further editing.

More Information

McFace Version 1.0 is sold as "partnerware": registered individual owners can resell copies for $20, keeping $10 and sending $10 to the author. The McFace master disk includes the McFace subroutine, sample programs, and documentation. McFace can be purchased by sending $25 directly to the author. The printed source code is also available for an additional $25 (all-FORTRAN, >2000 lines, > 500 Toolbox calls).

For further information, or to order the product, contact:

Dan Kampmeier
1310 N. Broadway
Urbana, IL 61801
(217) 328-5842

MacScheme and University Education

By Anne Hartheimer
Semantic Microsystems, Inc.

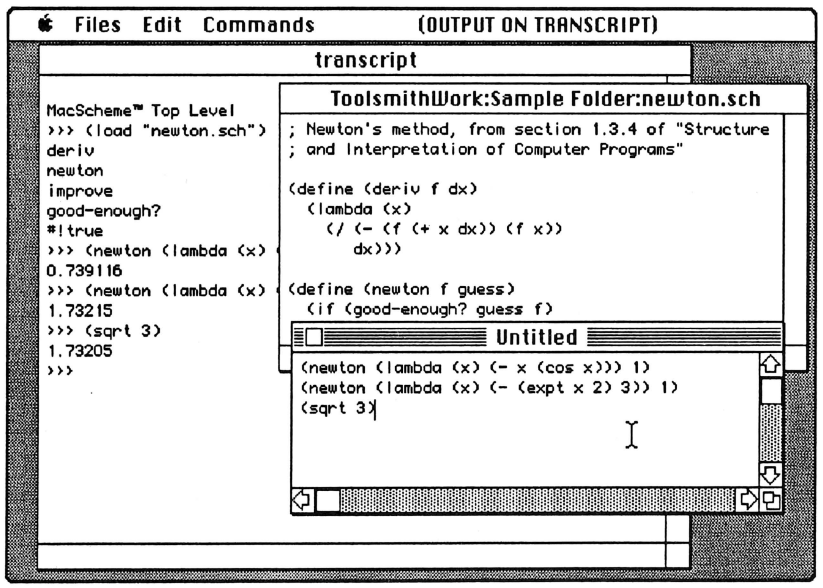
A revolution is taking place in introductory computer science courses, thanks to the textbook *Structure and Interpretation of Computer Programs,* by Hal Abelson and Gerry Sussman. Recent implementations of the Scheme programming language for popular microcomputers including the Macintosh have made this book accessible to a wide audience.

Structure and Interpretation of Computer Programs, which was developed for the introductory computer science course at MIT, organizes the syllabus around one crucial idea: control of complexity. Because the programming language used in the book is extremely powerful and easy to learn, the book is able to devote very little space to teaching it and is able instead to cover a very wide range of fundamental programming concepts.

To learn computer programming, one must learn to control complexity.

The programming language used in *Structure and Interpretation of Computer Programs* is a very simple, regular, and modern dialect of LISP known as Scheme. Since 1982, Scheme has seen heavy use at such schools as MIT, Yale, and Indiana University. With the 1985 advent of high-quality commercial implementations of Scheme running on the Macintosh and IBM PC, Scheme was quickly adopted by many other universities and colleges. Scheme is now used in courses on topics ranging from introductory computer science to operating systems, in compiler courses, and in graduate courses in artificial intelligence.

MacScheme by Semantic Microsystems was the first commercial implementation of Scheme for a microcomputer, and was indeed the first commercially available microcomputer implementation of any LISP with lexical scoping and first-class procedures (lexical closures; these are the hallmark of a modern LISP). The program runs on any Macintosh computer with at least 512K RAM, and Semantic Microsystems provides educational quantity prices as well as university site licenses.



A typical MacScheme screen

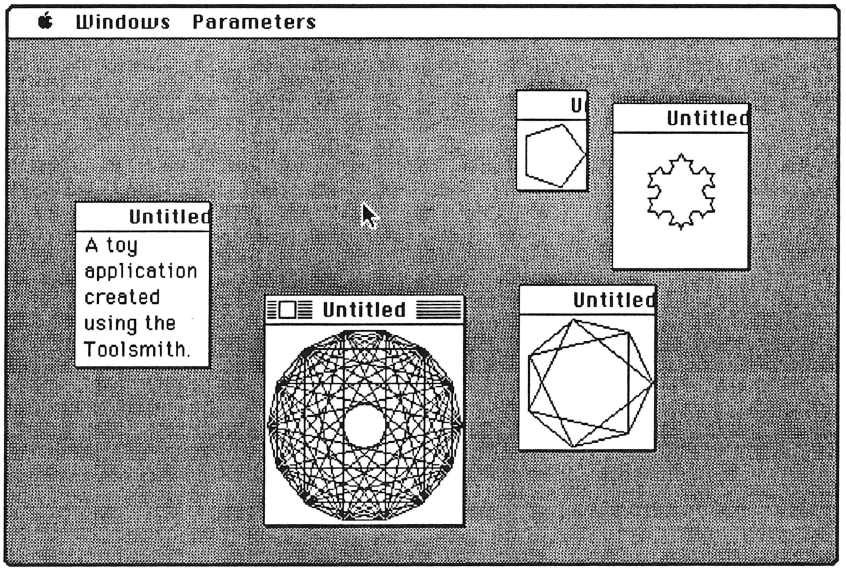
MacScheme combines an easy-to-use Macintosh-style interface with an interactive Scheme system, editor, source-code debugger, and tracer. Its editor helps the user with LISP syntax by highlighting matching parentheses and suggesting proper indentation. LISP features supported by MacScheme include lexically scoped variables, first-class procedures, macros, strings, vectors, arrays, and generic arithmetic, including both floating-point and unlimited-precision integer arithmetic.

The implementation is a compromise between an interpreter and a compiler; all code is compiled to byte code, which is then interpreted. This provides most of the speed of a compiler, while offering the interactive debugging features you'd expect only from an interpreter. MacScheme includes a Scheme-like subset of QuickDraw graphics commands. An enhanced version called MacScheme+Toolsmith, released in December 1986, provides high-level routines for easily creating menus and text-edit windows from Scheme, as well as providing Scheme access to the full set of Macintosh Toolbox traps.

More than 15 universities have used MacScheme.

More than 15 universities have used MacScheme for teaching, including Harvard, Yale, Boston College, the U.S. Coast Guard Academy, Carleton University (Canada), and the universities of Chicago, Rochester, Oregon, Utah, Amsterdam (Holland), and British Columbia.

A toy application created with MacScheme and Toolsmith.

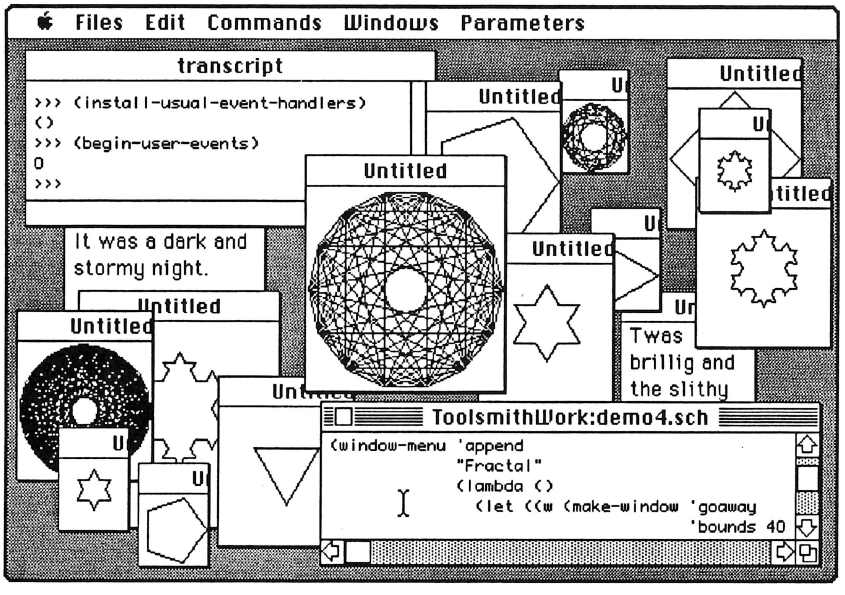


At the University of Chicago, MacScheme is used in a wide range of courses. The school's introductory computer science sequence teaches both LISP and Pascal, according to Professor Stuart Kurtz, who adds that "of course LISP nowadays means Scheme." At Chicago, students take two quarters to get through *Structure and Interpretation of Computer Programs*. The school also uses

MacScheme to teach programming languages and artificial intelligence, and in a course that introduces humanities students to mathematical thinking.

Scheme is unusual among LISP dialects in that it is standardized. The latest Scheme standard includes a denotational semantics for Scheme and (at the time this article was written) is to be published soon in ACM SIGPLAN Notices (see the list of references at the end of this article). Most Scheme programs will run without changes in both MacScheme on the Macintosh and Texas Instruments' PCScheme on the IBM PC. This works well for a school that has both Macintosh computers and IBM PCs.

MacScheme is unusual among LISP dialects.



This illustrates the interactive development mode of MacScheme and Toolsmith. After an application is debugged in this mode, you can enter an application mode and remove the MacScheme menus and the Transcript window.

For further information about MacScheme, contact:

Anne Hartheimer
Semantic Microsystems
4470 SW Hall #340
Beaverton, OR 97005
(503) 643-4539

References

Abelson, Harold and Gerald Jay Sussman with Julie Sussman, *Structure and Interpretation of Computer Programs*. Cambridge, Mass.: MIT Press, 1985.

Forbus, Kenneth, review of *Structure and Interpretation of Computer Programs*. *AI Journal*, Vol. 27, No. 1, September 1985.

Kilov, Haim, review of *Structure and Interpretation of Computer Programs*. *Byte* , Vol. 11, No. 12, November 1986.

Rees, Jonathan and William Clinger [eds.], "Revised Report on the Algorithmic Language Scheme," to appear in *ACM SIGPLAN Notices*, December 1986.

University of Chicago Computer Science Department

The Department of Computer Science is one of the fastest growing departments at the University of Chicago. In the past year, the number of graduate students in the M.S. and Ph.D. programs has more than tripled, and enrollment in the department's undergraduate courses has increased correspondingly.

The department offers 10 sections of a two-quarter introductory programming course (CS 103-104) that uses Macintosh Pascal. Students without a substantial mathematics background often take this course in order to gain a fundamental understanding of programming concepts (top-down design, algorithms and problem-solving, and so on) without having to spend large amounts of time learning to use a mainframe. The course has been very successful in attracting humanities and social science undergraduates.

Computer Programming as a Liberal Art (CS 110) also uses Macintosh Pascal. This course is oriented especially toward humanities and liberal arts students. Its purpose is to teach students how to think about the meaning of programming techniques and technology reflectively, as well as to teach them to program.

The department is using MacScheme in several of its more advanced courses, including CS 107 (an introduction to artificial intelligence and further programming principles), and CS 221 (programming languages). It is also in the process of designing other introductory programming courses that will use MacScheme (a statically scoped, LISP-like language) as the first computer language taught (rather than Pascal).

The department has a Macintosh laboratory, which provides undergraduates with access to hard disks and LaserWriter printers. Since the department is committed to UNIX and UNIX-based machines for its general-purpose computing and research, it is evaluating local area network file servers that can connect the laboratory Macintosh systems and LaserWriter printers (over an AppleTalk network) and can provide a gateway to the departmental and campuswide Ethernet system.

Future research will focus on developing systems to improve programming language instruction, based on artificial intelligence techniques. As hardware becomes ever more powerful, more sophisticated and more varied instruction environments are possible.

For further information, contact:

Don Crabb
Director of Instruction and Laboratories
University of Chicago Dept. of Computer Science
1100 East 58th St., Ryerson Hall
Chicago, IL 60637

Computer Science Applications Available from Kinko's

Kinko's Academic Courseware Exchange lists several applications that might be of interest to those in the computer science area. Brief summaries are provided here. A complete catalog of all of the courseware is available from Kinko's, at 4141 State Street, Santa Barbara, CA 93110; 800-235-6919. (Or try your local Kinko's outlet.)

Binary Trees (Version 2.3)

This is a program that allows you to experiment with binary trees in a graphical, interactive fashion. Minimum requirement is a Macintosh 128K, with any version of the Finder. Price: $7.

You can experiment with binary trees.

Event Tutor (Version 1.1)

This is a program to help in learning about "events," a concept that is integral to the Macintosh programming environment. Minimum requirement is a Macintosh 128K, with any version of the Finder. Price: $7.

Learn about Macintosh events.

Skel (Version 2.4)

This is a skeleton Macintosh program (written in Lisa Pascal) with a source code that can be studied and modified. Its purpose is to teach Macintosh program development. Minimum requirement is a Macintosh 128K, with any version of the Finder. Price: $7.

Examine sample Macintosh programs.

ForthTalk Kernel (Version 1.0K)

The ForthTalk Kernel provides a complete interactive object-based programming environment on the Macintosh computer. Minimum requirement is a Macintosh 512K and MacForth K2.4. Price: $30.

Learn to program.

ForthTalk Library (Version 1.0L)

The ForthTalk Library is a collection of flavors (flexible subroutines) that supports the ForthTalk Kernel program. Minimum requirement is a Macintosh 512K, MacForth K2.4, and ForthTalk Kernel. Price: $25.

Pascal Pointers (Version 1.43)

This application interprets and displays pictorially the results of Pascal commands that use pointers and dynamic variables. Minimum requirement is a Macintosh 128K, with Finder 4.1. Price: $22.

Get some help with Pascal.

The SmallGol Compiler (Version 1.0)

Use an animated compiler.

This is an animated compiler that displays animated, dynamic pictures that reveal how a compiler works. Minimum requirement is a Macintosh 128K, with Finder 4.1. Price: $16.50.

Turing's World

Examine Turing machines.

This application introduces the concept of Turing machines by providing a graphical environment in which to build, debug, and run them. Minimum requirement is a Macintosh 128K, with Finder 4.1. Price: $15.

Computer Science Applications from Vendors

Some exciting applications relevant to computer science have been developed by third-party vendors. These applications include programming languages, tutorials, utilities, and expert systems. The following list is a sampling of the kinds of Macintosh applications that are available in the area of computer science.

Here are some exciting third-party applications.

][in a Mac 2.0
An Apple II Plus, IIc, and IIe emulation system
Computer Applications, Inc.
12813 Lindley Drive
Raleigh, NC 27614
(919) 846-1411

B-Tree Helper
DevHelper
A variety of Pascal subroutines
(M)agreeable Software, Inc.
5925 Magnolia Lane
Plymouth, MN 55442
(612) 559-1108

AELM-51 Assembler/Editor/Loader
RASCAL
A software-development system
Metaresearch, Inc.
1100 SE Woodward
Portland, OR 97202
(503) 232-1712

Anatool
A tool for structured systems analysis
Abvent
9903 Santa Monica Blvd.
Beverly Hills, CA 90912
(213) 659-5157

Aztec C68k-c
A software-development system
Manx Software Systems
P.O. Box 55
Shrewsbury, NJ 07701
800-221-0440

APL.68000
A version of APL for the Macintosh
Spencer Organization
P.O. Box 248
Westwood, NJ 07675
(201) 666-6011

CLR ToolLib
CLR MathStatLib
Libraries that extend BASIC
Clear Lake Research, Inc.
5615 Morningside, Suite 127
Houston, TX 77005
800-835-2246, Ext. 199

Consulair Linker
Consulair Utilities
C-tree for Consulair Mac-C
Mac C
Mac C Toolkit
Software-development assistance
Consulair Corp.
140 Campo Drive
Portola Valley, CA 94025
(415) 851-3272

ExperLisp 1.5
ExperLisp-Plus
ExperLisp Talker
ExperLogo 1.1
ExperOPS5
ExperOPS5-Plus
ExperProlog II
ExperFacts
ProLogo
Support for expert-systems development
ExperTelligence, Inc.
559 San Ysidro Drive
Santa Barbara, CA 93108
(805) 969-7874

C.P. Mac
CP/M program interpreter
Logique
30100 Town Center Drive
"O" Suite 198
Laguna Niguel, CA 92677
(714) 953-8985

Cut and Paste BASIC
Library of BASIC subprograms
Technisoft Engineering
P.O. Box 563
Bremerton, WA 98310
(206) 598-4177

DeSmet C
A software-development environment
C Ware Corp.
P.O. Box C
Sunnyvale, CA 94087
(408) 720-9696

dBx Translation System
Translator
Desktop AI
1720 Post Rd. E. No. 3
Westport, CT 06880
(203) 255-3400

MacForth
MacForth 3D Library
MacForth Developer's Kit
MacForth Plus
Programmer's Toolkit
Software-development tools
Creative Solutions, Inc.
4701 Randolph Road, Suite 12
Rockville, MD 20852
(301) 984-0262

Mach-1 Multitasking Forth
Programming
Palo Alto Shipping Co.
P.O. Box 7430
Menlo Park, CA 94026
(415) 854-7994

HeapShow Version 3
Displays heaps
B/T Computing Corp.
P.O. Box 1465
Euless, TX 76039
800-843-6463

LightSource
Authoring system
Computer Learning Systems, Inc.
RFD #8 Box 375
Concord, NH 03301
(603) 783-4708

LightspeedC
A C language compiler
Think Technologies, Inc.
420 Bedford Street
Lexington, MA 02173
(617) 863-5590

Logic Compiler
An aid for designers of hardware
Capilano Computing Systems, Ltd.
P.O. Box 86971
N. Vancouver, BC
Canada V7L 4P6
(604) 669-6343

Logo
Microsoft BASIC 2.1
BASIC (interpreter)
Microsoft FORTRAN 2.1
Programming languages
Microsoft Corporation
16011 NE 36th Way
Box 97017
Redmond, WA 98073
(206) 882-8080

MacExpress
Utility
ALSoft, Inc.
P.O. Box 927
Spring, TX 77383
(713) 353-4090

MacAdvantage
Programming language
Pecan Software Systems, Inc.
1410 39th Street
Brooklyn, NY 11218
(718) 851-3100

MacASM Macroassembler
Development system
Mainstay
28611B Canwood Street
Agoura Hills, CA 91301
(818) 991-6540

Macintosh Pascal
Macintosh 68000 Development
Programming systems
Apple Computer, Inc.
20525 Mariani Avenue
Cupertino, CA 95014
(408) 996-1010

MacLanguage Series Pascal
MacLanguage Series Database ToolKit
Pascal tools
TML Systems
P.O. Box 361626
Melbourne, FL 32936
(305) 242-1873

MacMeth
Modula-2 language
Modula Corp.
950 N. University Ave.
Provo, UT 84604
800-545-4842

MacMumps
Programming language
MGlobal
1601 Westheimer, Suite 201
Houston, TX 77006
800-257-8052

MacNosy
Disassembler
Jasik Designs
343 Trenton Way
Menlo Park, CA 94026
(415) 322-1386

Neon 1.5
Neon Assembler
Programming language
Kriya Systems, Inc.
6 Export Drive
Sterling, VA 22170
800-345-7492

MacPlot Graphics Library
FORTRAN package
Lipa Software
165 Harcross Road
Woodside, CA 94026

MacScheme
Programming in LISP
Semantic Microsystems
4470 SW Hall Street, Suite 340
Beaverton, OR 97005
(503) 643-4539

MacProlog
AI language
Programming Logic Systems, Inc.
31 Crescent Drive
Milford, CT 06460
(203) 877-7988

MasterFORTH
Programming language
MicroMotion
8726 S. Sepulveda Blvd., No. A171
Los Angeles, CA 90045
(213) 821-4340

McAssembly
68000 development
Signature Software
2151 Brown Avenue
Bensalem, PA 19020
(215) 639-8764

Megamax C
C language development
Megamax, Inc.
P.O. Box 851521
Richardson, TX 75085
(214) 987-4931

On Stage
Pascal compiler
Step-Lively Software, Inc.
622 Watervliet-Shaker Road
Latham, NY 12110
(518) 785-7214

Pascal Extender
C Extender
Utilities
Invention Software Corp.
P.O. Box 3168
Ann Arbor, MI 48106
(313) 996-8108

PCMacBasic
Compiler
Pterodactyl Software
200 Bolinas Road, No. 27
Fairfax, CA 94930
(415) 485-0714

PortaAPL
APL interpreter
Portable Software
60 Aberdeen Avenue
Cambridge, MA 02138
(617) 547-2918

Talkshow
Smalltalk subset
Rune Enterprises
1502 Kirkham
San Francisco, CA 94122
(415) 681-1763

Project C
Programming language
MAComb Projects
129 N. Randolph
Macomb, IL 61455
800-851-2753

Prolog II
AI language
Avenue Software, Inc.
1173 West Charest Blvd., Suite 250
Quebec City
Province of Quebec, Canada
G1N 2C9
(418) 682-3088

P-tral
BASIC-to-Pascal translation
Woodchuck Industries, Inc.
340 W. 17th Street, Suite 2B
New York, NY 10011
(212) 206-6490

QUED
Editor for developers
Paragon Courseware
4954 Sun Valley Road
Del Mar, CA 92014
(619) 481-1477

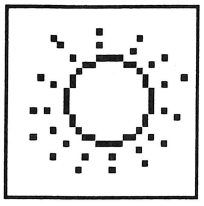
Softworks BASIC
Softworks C
Programming languages
Softworks, LTD
2944 N. Broadway
Chicago, IL 60657
(312) 975-4030

VideoWorks
Animation-development system
MacroMind, Inc.
1028 W. Wolfram Street
Chicago, IL 60657
(312) 327-5821

ZBasic
BASIC compiler
Zedcor, Inc.
4500 E. Speedway, Suite 93
Tucson, AZ 85712
800-482-4567

True BASIC
BASIC programming
True BASIC, Inc.
39 S. Main Street
Hanover, NH 03755
(603) 643-3882

Spotlight



The **Spotlight** section will focus on some of the Macintosh activities that are going on at various schools and universities. Typically, there will be two parts: the first will describe the activities that coincide with the theme of that particular *Wheels for the Mind* issue; the second will be more general, and describe a variety of interesting programs and activities.

University of North Carolina, Chapel Hill: Introduction to Computers Course

By Dr. Peter Calingaert, Professor of Computer Science, and Susan Gauch, Teaching Assistant

This fall, Professor Calingaert has incorporated a Macintosh section into his Introduction to Computers course at UNC. For the first time, students will be able to choose between using a Macintosh and using an IBM PC.

Half of the course is conceptual and half is practical.

The course really consists of two subcourses. One is conceptual—the study of the technical and social aspects of computing. The other is practical, and involves hands-on sessions at the computer labs around campus, primarily in Venable Hall.

Why was the Macintosh computer chosen?

Reasons for using Macintosh in the course include:

- The user interface is consistent from application to application, which makes it easier for students to learn quickly.
- The user interface is attractive to students.
- Macintosh has already been used successfully to teach Pascal programming with Macintosh Pascal.

Software to be used in the introductory course will be the Guided Tours of Macintosh and MacWrite™, Microsoft Excel, and some form of Logo. Other pieces of software are under discussion.

Calingaert and Gauch feel there really is a difference in the ease with which material is picked up using Macintosh. They believe that students will be extremely enthusiastic about being able to use the Macintosh during this term.

For further information, contact:

Dr. Peter Calingaert
Department of Computer Science
University of North Carolina, Chapel Hill
(919) 962-3970

MIT: LISP Application

Research is under way on the use of artificial intelligence in music.

Research and development is under way at the Massachusetts Institute of Technology in the area of artificial intelligence and its applications in music. The Entertainment Group at the MIT Media Laboratory has been developing Macintosh software for over a year, focusing on dramatic "real-time" music and animation projects, and on user-interface elements.

David Levitt, in his graduate research, worked with LISP and musical style representations. His work involved "talking" in LISP to express stylistic characteristics of music. "Templates" were to be developed that would call up the proper style (for example, bass player, bebop, ragtime, New Orleans) to produce improvisations and arrangements. The template would determine how an arrangement would be constructed from a given melody and/or chord progression. These templates could be refined into something similar to a spreadsheet template, enabling arrangers, composers, and conductors to call up various styles without having to understand artificial intelligence—or even computers, for that matter.

A novice can invent a new control panel for a real-time musical instrument.

Another result of this research is a Macintosh application called "Hook Up: An Iconic, Real-Time Data-Flow Language for Entertainment," which was presented last year at MacWorld Expo '86 and SIGgraph '86. With this tool, a novice can invent a new control panel for a real-time musical instrument by drawing a diagram on the Macintosh screen—without ever referring to a manual. The same toolkit can be used to control real-time animation. In brief, Hook Up is a graphic programming language in which black lines represent data equipotentials (wires) and icons represent functions and input/output devices. Music and sprite icons allow a user to learn the language by constructing many simple entertainment devices.

Work is under way to port music software from a LISP machine to Macintosh using Common LISP from Coral Software (*unreleased at time of writing*) and to define mouse and window interfaces. A significant milestone was reached in late October when this work resulted in the ability to use Macintosh to "talk" to a MIDI

synthesizer in LISP. These exciting developments are just the beginning of many possibilities that will be realized as new and more powerful hardware evolves.

For further information, contact:

David Levitt
MIT Media Lab
Massachusetts Institute of Technology
20 Ames St.
Cambridge, MA 02139

University of Utah: Computers in Liberal Education

Macintosh is being used in many different areas of computer science at the University of Utah. Most programming courses are taught using Macintosh, including classes in Pascal and C. Presently, Pascal is being taught using Macintosh Pascal; future courses will incorporate Lightspeed Pascal from Think Technologies. Courses in C presently use Lightspeed C, also from Think Technologies.

Pascal, LISP, and C are all being taught.

Another language in use at Utah is LISP. The university has derived its own version of LISP, called Portable Standard Lisp (PSL). It has been ported to several machines, including Macintosh, and is being used for problems in artificial intelligence, graphics, and computer science. As technology and machine capabilities increase, PSL will be even easier to use for programming-language instruction.

One course teaches how the intellectual content of the computing discipline relates to other disciplines.

Professor Bill Salmon teaches a course called "The Computer Age: Symbols, Logic, Machines," which is an experimental course within the university's liberal education program. (All students must complete a required number of hours of courses in physical sciences, humanities, fine arts, and social sciences in order to satisfy liberal education requirements.) Professor Salmon's course is designed to teach students how the intellectual content of the computing discipline relates to other disciplines and how the computer is really a logic machine—acting out the laws of logic. This logic is also present in all other disciplines, and as Professor Salmon says, "This course is a connections course. We want to see the connections between the laws of logic resident in computers and the laws of logic in other fields."

One example of such connections is in the field of history. The course examines the ideas of mechanical implementation of logic, how they came about, and how they are embodied in the computer.

Students have three lecture hours of class per week along with four lab hours. In the lab, they are introduced to the Macintosh with the

Students work with MacWrite, Multiplan, and Logo.

textbook *Approaching Macintosh* (Addison-Wesley) and MacPaint software. Next, students learn MacWrite, which they use for writing term papers. They then learn a conventional use of a Multiplan spreadsheet. After these applications are learned, Multiplan is used in a nonconventional way to teach logic. During the final four weeks of the course, students receive an introduction to programming using Microsoft Logo. Logo was chosen because it is easy to learn and allows flexibility in programming. The students finish the course by doing a little bit of artificial-intelligence programming.

In just one course, students receive a thorough introduction to the philosophy of computing, its relation to other disciplines, actual hands-on experience with several applications, and an introduction to programming. No matter what disciplines these students go on to study, they start with a very good feel for the role of computing in today's world.

For further information, contact:

Computer Science Department, MEB 3406
University of Utah
Salt Lake City, UT 84112

University of Waterloo: New Vistas in Computer Literacy

By S.L. Fenton, J.P. Black, and V.A. Dyck
Department of Computer Science, University of Waterloo

The philosophy is to provide a strong background in hardware and programming in first-level computing courses.

The University of Waterloo has been intimately involved with the use of computers by students for over two decades. Throughout most of this period, its philosophy has been to provide a strong background in hardware and programming in first-level computing courses so that the students could use computers effectively. Computer literacy was equated in many ways with a knowledge of programming, even for students with no intention of pursuing a specialty in computer science. The recent advances in hardware, and the development of relatively easy-to-use software, have opened the world of computing to an expanding group of people who neither require nor desire an in-depth knowledge of traditional computer science concepts. What is needed is an understanding of how and why computers play an increasingly important role in their lives.

A revolutionary approach to teaching this type of computer literacy is required. At the University of Waterloo, discussions with representatives from numerous disciplines led to the creation of a new course called CS 100: Introduction to Computer Usage.

The ambitious goal of CS 100 is to provide students with the knowledge and skills needed to use the computer as a tool for solving problems that they will encounter throughout their personal, academic, and working lives. The Macintosh was chosen as the ideal vehicle for implementing this fresh approach to computer literacy, because of the way it simplifies the world of computing.

The intent is to provide students with the tools they will need for solving problems.

Design of CS 100

In CS 100, students develop a broad understanding of computers: what they are and what they can do. The focus of the course is on universal computer tools such as word processing, electronic spreadsheets, data-base management, graphics, and communications, so that students become able to take advantage of the computer for many of their day-to-day tasks. The course concludes with a very brief introduction to programming, as a bridge to possible subsequent programming courses.

Two hours of lectures per week are supplemented by extensive practical experience with a personal computer. The emphasis in the lectures is on general techniques and principles embodied by a number of commercially available programs. The laboratory periods provide the students with exposure to a particular set of those computer tools.

A two-week section (or four lectures) is allotted to each computer tool. The objectives of each section are as follows:

Four lectures are allotted to each computer tool.

- To teach the basic concepts of the tool.
- To cover features commonly provided.
- To describe any advanced features.
- To place the tool in a wider perspective.

To provide additional motivation, a number of diverse applications of each tool are woven into the lecture sequence. For example, the section on text processing begins with lectures on the basic, common, and advanced features of word processors, as illustrated by several programs (MacWrite, PFS:File, and WordStar). The section concludes with a tour of the broader world of text processing, exploring exciting new applications and research areas such as idea processors, dictionary automation, text analysis, concordances, and electronic publishing.

Each lab consists of two parts: a detailed tutorial guiding the student through particular tasks, followed by an assignment that exercises the skills that the student has acquired. Periodic lab quizzes are given to evaluate the student's skills with these tools. Midterm and end-of-term exams test the student's understanding of the general concepts.

Labs include tutorials and assignments.

One of the key requirements of the course is that the hardware and software chosen provide a computing environment that is as easy to use as possible. Not only are most of the students completely

Macintosh Pascal was used as the introductory programming language.

unfamiliar with computers, but many of them feel ill at ease with, if not threatened by, the prospect of facing one. The Macintosh, with its consistent and easy-to-use graphical interface, was a natural choice to support the course; it was complemented by high-quality software from Apple, Microsoft, and Eqtron for most of the applications covered. MacWrite, MacPaint, Multiplan, Chart, File, and MacBase were selected to illustrate the application areas; Macintosh Pascal was used as the introductory programming language.

Waterloo and MacJANET

The need to be able to share software was a major requirement.

Because of financial and facility limitations, enrollment in CS 100 is currently restricted to 600 students per term, with more clamoring to get in. Managing the physical and human resources for such a large course presents some unique challenges. For example, the need to share software, data files, and peripherals makes the provision of some sort of distributed system or networking software a major requirement.

A user-friendly network was developed to integrate a number of Macintosh workstations.

The University of Waterloo had already developed several networks to accommodate student computing needs. A user-friendly network was developed to integrate a number of Macintosh workstations with a Macintosh network server. This educational network, Waterloo MacJANET, controls the file space for all the course software and data, and for student files. As a result, problems of managing program and data disks are solved, and software copyright restrictions can be enforced.

Summary

CS 100 is an important step in the general movement at the University of Waterloo toward providing student computing for *all* students on campus. In the first year alone, the course was taught to approximately 1,500 students. The response to CS 100 has been enthusiastic. Students who never wanted to touch a computer—much less use one in everyday life—are requesting access to these tools for all their assignments. Indeed, using the computer as a *general tool* now has meaning for a growing number of students on campus.

The authors wish to acknowledge the generous support of Apple Canada Incorporated, Microsoft Canada Incorporated, Eqtron Corporation, and Computer Systems Groups at the University of Waterloo. Without their help, we would have had significant difficulties in developing CS 100.

S. L. Fenton is Manager of Project Development, Office of the Dean of Computing and Communications; J.P. Black is an assistant professor in the Department of Computer Science; and V.A. Dyck is an associate professor in the Department of Computer Science; all at the University of Waterloo.

Princeton University: Algorithm Visualization Tool

Princeton University is conducting ongoing research on development of a visualization tool called MacBalsa. "Balsa" stands for *Brown University Algorithm Simulator and Animator*, and as can be deduced from the title, was originally developed at Brown University. Professor Robert Sedgewick and then-graduate student Marc H. Brown ported Balsa to Macintosh from a UNIX environment, and further work is now under way at Princeton to extend the system and add capabilities.

Animating algorithms allows the user to see what an algorithm is doing.

MacBalsa is an integrated software environment designed to "animate" algorithms—in other words, to show what an algorithm is "doing." It allows users to construct "movies" for illustrating or learning about properties of programs. The subsystems within MacBalsa are designed to be used for algorithm animation while emphasizing performance and a high-quality user interface. MacBalsa allows a prepared script; a student can run it interactively, or it can be used on a shared system with the instructor stepping the class through a lecture.

The three main uses for MacBalsa are teaching, research, and systems programming. In teaching and research, the most natural application is to dynamically simulate the execution of algorithms.

Balsa was originally used as a teaching tool at Brown University in 1983 and 1984. During the lecture portion of one course, all students were Balsa users, and the instructor could present material dynamically. Each student's workstation would replay a "script" of material that had been previously created and saved. At key points, the system would pause and then resume when the student was signaled by the instructor. The instructor usually based the lecture on a running commentary of the simulation. Lecture scripts were available for students to play back later, and provided a starting point for them to further interact with the material in their own ways.

The simulations in the course covered fundamental areas of computer science.

The text and simulations in the course covered basic algorithms in fundamental areas of computer science: mathematical methods, sorting, searching, strings, geometry, and graphs, as well as areas such as algorithm machines and dynamic and linear programming. In preparing the teaching material, two important tasks were involved: conceptualizing a graphical representation of the program that would convey the essence of the algorithm and its data structures, and creating a good script that could be integrated into a lecture. It took between 15 and 25 hours to program the various algorithms and views and an hour or two to develop a script. Once formulated, however, these materials could be used for teaching and research applications, and by the end of the semester, a "dynamic book" had been developed to accompany the text, complete with a graphic table of contents composed of icons from the various

animations. By selecting an icon, the user can evoke the corresponding lecture script for playback or interaction.

This dynamic book creates a basis for further research applications so that the dynamic interactive graphics used in the classroom can also be used in research. Already, Balsa has been valuable in the design and analysis of graph-searching algorithms, sorting algorithms, and priority-queue algorithms.

There are applications in other disciplines as well.

As hardware and software products evolve, more will be done to extend and add capabilities to MacBalsa. The environment presently has a great many uses in computer science and has the possibility for extension into other disciplines, as well.

Information for this article comes from IEEE Software *magazine, January 1985; and from Phillip Thompson, Manager of Advanced Technology in Computing at Princeton University. Many thanks to him and to Professor Robert Sedgewick, Chair, Department of Computer Science, Princeton University.*

University of Minnesota: Pascal Procedures

MEWS stands for Menu, Event, and Window System.

The Microcomputer Systems Group at the University of Minnesota has developed a set of Pascal procedures to be used with the TML Pascal compiler for Macintosh. It's called MEWS, which stands for Menu, Event, and Window System. Although most programmers using MEWS will be familiar with TML Pascal, it is also possible to use the MEWS library with Consulair's C, Apple's 68000 Development System, and other languages that use the Consulair linker. (It may be necessary to write your own interface to MEWS procedures, however, if you are not using TML Pascal.)

Programmers are provided with a general-purpose template for writing Macintosh programs.

MEWS provides programmers with a general-purpose template for writing Macintosh programs. Rather than spending the bulk of time writing code to implement the Macintosh user interface, programmers can spend time on the code that is unique to their specific applications.

The MEWS library handles:

- Initializing all the Macintosh Toolbox managers.
- Creating menus and menu items and the Desk Accessory menu.
- Creating windows with optional grow boxes and scroll bars.
- Handling events in application and System windows.
- Handling window resizing, scrolling, and updating.

- Handling clicks, double-clicks, and drags.
- Handling out-of-memory conditions.

While many of these tasks seem simple, there are many reasons why you should use MEWS to handle them:

MEWS provides a standard way to work with menus and windows.

MEWS provides a simple standard way to create and manipulate menus and windows. Compare MEWS with all of the possible (and confusing) ways of handling menus and windows consistently through Toolbox calls. By using MEWS, you can make all of your windows behave consistently.

MEWS handles some difficult tasks, such as servicing scroll bar events and doing the actual window scrolling. The code to do this can be tricky to write (and rewrite) for each program window.

MEWS handles many of the *backdoor* paths through the event manager, such as desk accessory and modeless dialog events. It is easy to ignore or misdirect these events if you try to handle them yourself.

Finally, MEWS gives you a scapegoat if your program doesn't work properly.

MEWS operations can be divided into two main phases: *initialization* and *event driven.*

Initialization takes place first, and then the program enters the event-driven phase.

During *initialization*, your application is the driving force in the program's flow. Your program calls MEWS to initialize the Macintosh software, create menus, and open windows. The initialization phase usually happens only once, when you start up your application.

Once the initialization phase is complete, the program enters the *event-driven* phase. In this phase, user-generated events (such as mouse clicks, drags, and key presses) provide the driving force in the program's flow. During the event-driven phase, your application program loops, waiting for events to occur. MEWS detects these events and calls the proper procedure in your application (or in the system) to service the event. You write procedures (in your program) to service the events in which you are interested.

What does this mean to the programmer? It means that programs that use the MEWS library will first execute an initialization section and will then go into an event-driven mode. Once the program enters event-driven mode, MEWS calls one of the procedures you wrote to service an event *whenever the event occurs*. After your procedure has serviced the event, MEWS continues to look for events and call appropriate event-service routines.

For further information, contact:

Microcomputer Systems Group
University of Minnesota
Room 125, Shepherd Labs
100 Union St., SE
Minneapolis, MN 55455

Information for this article was excerpted from the MEWS version 2.1 documentation,
© 1986 by the University of Minnesota.

Boston College: Computer Science Courses

At Boston College, approximately 800 undergraduate students take an introductory computer science course using the Macintosh.

A public facility with 150 Macintosh computers is available to students.

Faculty members have their own microcomputer lab.

A public facility with 150 Macintosh computers is available to students, a Macintosh Microcomputer Resource Center has been set up for faculty use, a laboratory of Macintosh computers is utilized by the School of Management, and each faculty member in the computer science department has his or her own Macintosh computer.

The following courses have involved heavy use of the Macintosh computer:

- Pascal Programming
- Software Development on the Macintosh
- Assembly-Language Programming
- BASIC Programming

Several other courses in the undergraduate computer sciences major are being modified to include the use of the Macintosh computer. In addition, Boston College will be initiating a master's program in computer science in the near future and has plans to use the Macintosh computer product line in support of courses in artificial intelligence, expert systems, and a variety of new graduate-level courses.

Virginia Tech: Macintosh and UNIX

During the spring of 1983, the faculty of the Department of Computer Science voted to institute a requirement of personal computers for majors. This would assure students of better access to computing facilities, as the university's public facilities were fast becoming overcrowded. Students would also have the best opportunity to learn to use computers as tools, for both computer science and other coursework.

Computer science majors are required to have their own personal computer.

The selection of machine was made early in 1985, and the machine chosen was the Macintosh XL. Software used is UNIX System V. Assemblers and compilers for C, FORTRAN, and Pascal were provided by UniPress Software, Inc. as the distributor for several other companies. The department made a separate purchase of UNIX System V (source code) for Motorola 68000 processors from Western Electric. In the fall of 1985, more than 230 freshmen majoring in computer science at Virginia Tech acquired microcomputers running the UNIX operating system. Purchases have also been made by other students, faculty, and staff.

There were several reasons the department selected the UNIX operating system for training computer science majors:

The UNIX operating system was chosen as the operating system for training computer science majors.

- *A familiarity with UNIX will directly benefit students entering the job market.* UNIX is becoming a standard operating system for micros, minis, and large systems, and it is supported by leading computer manufacturers. Because of this emerging standard, there is a tremendous demand for people who are experienced in UNIX and C. On campus, there was a demand by students for courses in these areas.

- *Knowing C and UNIX facilitates learning of computer science methods (and vice versa).* Many aspects of this language and operating system lend themselves to learning methods of approaching problems in computer science. Students learn to use ideas that are often taught only in the abstract as formal theory; working with the file system helps them develop an understanding of tree structures and hierarchies; structured programming with the shell helps teach good software development practices; C is useful for systems programming because of its handling of data types, its use of pointers, its ability to manipulate bits, and its storage flexibility, expressed in part through unions; UNIX tools are useful in compiler courses and enable students to apply what they have learned for useful tasks; UNIX text-processing software increases student exposure to handling textual data in elegant ways; the style of UNIX development is important to learn as an example of good software engineering; and students become system administrators and learn even more about operating-system support functions and handling different devices.

- *Using UNIX and C allows interchange of software developed in research projects.* All major software research and development efforts of the computer science department are being done with C and/or UNIX. This enables results of research to be given to students for use in other projects.

- *UNIX software is relatively complete.* Students need not purchase several expensive packages. UNIX system software is fairly costly, but it does provide a large number of services such as other languages (C, Snobol), games, text-processing routines, source-code maintenance facilities, and macroprocessors.

Pascal was used in the introductory course.

The microcomputer requirement in the department appears to have actually encouraged enrollment, since the 1985–86 freshman class has 60 more students than previous classes. The first introductory computer science courses are taught using Pascal. Whereas in the past, students used Pascal compilers on VAX 11/780 mainframes, they can now compute on their own Macintosh systems. The 68000 assembler is being taught in the third introductory course. Each quarter, the department plans to add at least one more course to the curriculum to be used with the Macintosh XL and UNIX System V. All graduate students have the opportunity to take UNIX and C courses, as well.

In addition, there are a number of research projects under way at Virginia Tech that utilize UNIX on microcomputers. One area involves information retrieval and the adaptation of the SMART information retrieval system to microcomputers for noncommercial purposes. A microcomputer version that will include the most important functions of SMART is being worked on presently. The goal of the Microcomputer Information Retrieval (MIR) project is to focus on automatic indexing of documents and queries. It is expected that MIR will be used to maintain various data bases that need to be searched regularly.

Another project was initiated by computer science students. In this project, they have drawn up requirements, specifications, and design documents for the Extended Library System Interface (ELSI). The ELSI package allows a microcomputer to serve as a front end to the million-entry on-line library catalog. This makes access simpler and will serve as a vehicle for experimentation in future years.

These exciting projects are early examples of ways that students can use the UNIX operating system in conjunction with a personal computer. It will be interesting to see what can be accomplished as they learn to better utilize these advanced tools.

Some information for this article was taken from an article written for Microcomputers for Information Management, *March 1986, titled "UNIX Micros for Students Majoring in Computer Science and Personal Information Retrieval." Many thanks also to Professor Edward Fox and Sandra Birch of the computer science department at Virginia Tech.*

University of Illinois Champaign/Urbana: Cray Finder

The main goal of this project is to turn the Cray into a coprocessor for the Macintosh, providing the power of the Cray and the ease of use of the Macintosh. Programs will run on the Cray using the standard Macintosh convention of selecting items with single mouse clicks and activating items by double-clicking on an icon, making the involvement of the Cray transparent to the user. The Macintosh program performing this function will be called the Illinois Cray Finder.

The main goal of this project is to turn the Cray into a coprocessor for the Macintosh.

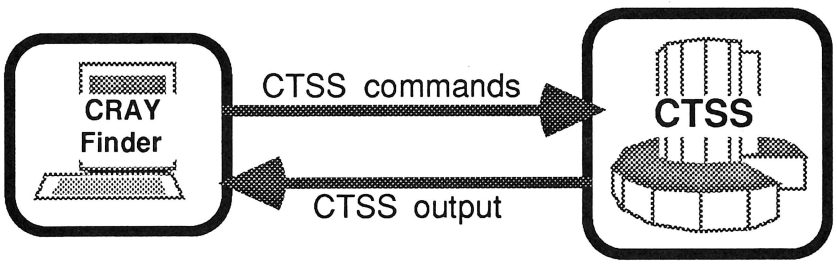
The following is a description of the steps required to achieve the goal of "hiding" the Cray:

Overall Design Considerations and System Requirements

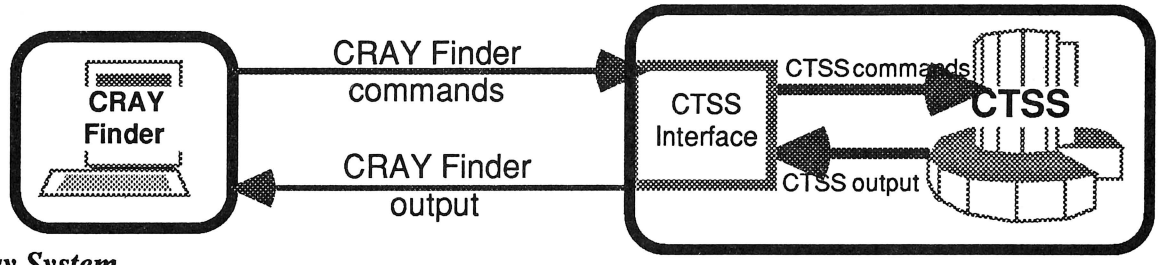
The basic idea is to have the Macintosh interpret all the information coming from the Cray, reformat it to appear in a Macintosh environment, and use that same environment to issue commands to the Cray. There are two basic approaches to programming this interface: the "unmodified Cray" philosophy and the "smart Cray" philosophy.

In the unmodified Cray system, the mainframe system doesn't know it is communicating with the Cray Finder: the commands to the Cray are identical to those a user would enter. This system has the advantage of requiring no mainframe programming. It has the disadvantages of inefficient use of network time and resources, and difficulty in providing standard condition processing.

Unmodified Cray system



In the smart Cray system, the Cray Finder communicates with a corresponding interface program running on the Cray, which then issues the actual Cray commands. This scheme makes efficient use of the network. For example, instead of issuing the command to transfer a file from the VAX to the Cray X-MP, which would entail sending more than 200 characters, a two-character code could be sent to the Cray Finder, producing the same result but using substantially less network time. This scheme is somewhat less flexible than the unmodified Cray scheme and requires more mainframe programming expertise.



A smart Cray System

It is not now clear which method is better. The MacWorkstation modules available from Apple Computer clearly have chosen the smart Cray method (which is not available for CTSS). We will experiment with both systems. Luckily, the choice of the scheme will have no effect on the design of the user interface.

Currently, a user can access the Cray by using a modem and a phone line.

Hiding the Cray also means hiding the network. Currently, a user can access the Cray from the Macintosh by using a modem and a phone line, by using the campus Sytek network, or by using the AppleTalk-Ethernet bridge to connect directly to the VAX front end. On startup, the Cray Finder will ask the user which connection system will be used. From this point on, the program will act identically regardless of whether the connection is remote and low speed, or local and high speed.

To put the user in a familiar environment, the Cray Finder will function like the Macintosh Finder program. File transfer will be accomplished by moving icons, and Cray programs will appear as icons in the Cray window.

Communications Modules

How to talk to a supercomputer.

The Cray Finder will be connected to the VAX front end in one of four ways: dialup, direct serial connection, AppleTalk-to-Ethernet, or direct Ethernet.

Dialup: This is the most familiar connection scheme. We would like to support all modem speeds from 300 to 9600 bps and have the system usable at all speeds. In addition, we would also like to support the industry-standard modem protocol, so that connections can be made automatically.

Direct serial connection: The current direct serial connection network on the UIUC campus is Sytek, which runs at 9600 bps. This method of communicating with the Cray system will only be used on the UIUC campus.

AppleTalk-to-Ethernet: Using this connection, the Cray Finder communicates with the VAX front end by sending packets over AppleTalk that are picked up by the Kinetics Ethernet gateway and are then dumped on the local Ethernet network. While on Ethernet, the TCP/IP protocol is used. We want this module to look to the Cray Finder as a very high-speed serial port. This system will initially only be used in the Interdisciplinary Research Center (IRC),

but may eventually also be used remotely (for example, over the Vitalink satellite network).

Direct Ethernet: A hardware interface (not yet on the market) will connect the Macintosh SCSI port directly to Ethernet. Again, once on Ethernet, the TCP/IP protocol must be used. This connection scheme will be virtually identical to the AppleTalk-to-Ethernet bridge, except that much higher speeds are possible and the AppleTalk network is not bogged down.

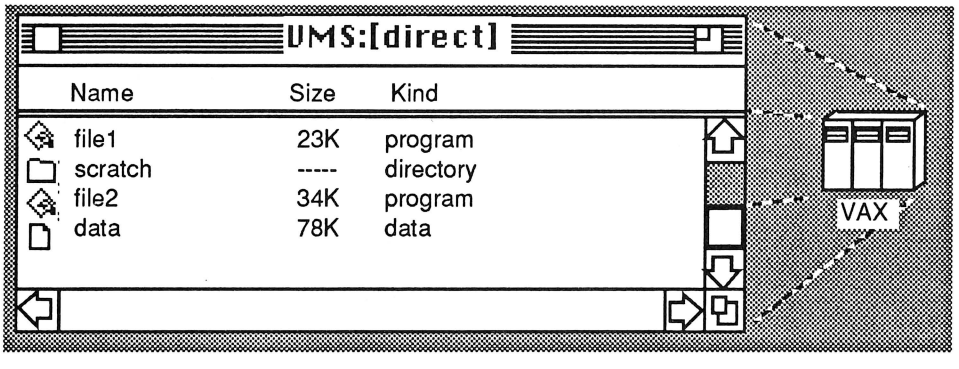
As stated earlier, these communications modules should all be called identically, so the Cray Finder does not need to know which communications path is being used. For remote low-speed users, efficiency of file transfer is crucial. Any data compression techniques developed at NCSA will be promptly incorporated into these communications modules.

Of course, the sign-on procedure to the VAX and/or the Cray will be completely automated.

File Transfer Modules

The first and most obvious task is to implement file transfer, such that, to the user, each mainframe module (VAX, Cray, CFS) looks like a Macintosh disk. As with the Macintosh, files will be selected and then dragged to the new disk. Directory listings will be text only, similar to the listing of files in the Macintosh Finder when the View by Name command is selected. In the future, the directory listing may have an option for iconic representation of files, with different types of files on the remote computers having different icons.

In the future, icons may be used to represent files on different computers.



Mainframe file directories

Subdirectories will be listed as folders in the directory listing, for VMS, UNIX, and CFS. Selecting and opening a folder/directory will open it into another window, just as when using the Macintosh Finder. The name of the window will be the corresponding path name.

Mainframe File Directories in the Cray Finder

We want identical directory listings for at least five different systems: Macintosh, VMS, UNIX, CTSS, and CFS. In addition to this generic directory listing form (name, size, date, type), a more informative directory listing will be available, giving system-specific information.

File transfers will be done in the same way they are now done on the Macintosh computer.

As stated above, file transfers will be specified in the same way that transfers are specified in the Finder: files are highlighted and then dragged to a new "disk." The default for transfers is to have the files named the same on the destination computer as they are named on the source computer. This will not always be possible. If the destination already has a file with that name, a dialog box identical to the Finder dialog will ask for confirmation. If the destination machine has a maximum file name size smaller than the source, the name may be truncated. Whether the name is truncated or there is a prompt for a new name will be set under a Preferences menu in the Cray Finder.

File transfers will be of ASCII text only, with one exception. The Cray Finder will be aware of the LF/CR preferences of the various machines and will convert appropriately. Transfer from the Macintosh to other machines will be via Kermit. Transfer from the VAX to CTSS will be via MOVE and the associated utilities. Transfer to CFS will be via the CFS utilities.

The one exception will be Macintosh applications (and data). These files will be "BinHexed" before transfer, which cleans them up to look like ASCII text. They can then be converted back to the Macintosh applications through the public domain utility BinHex 4.0. This will enable NCSA to exchange Macintosh applications and formatted data with remote users, via the front-end VAX.

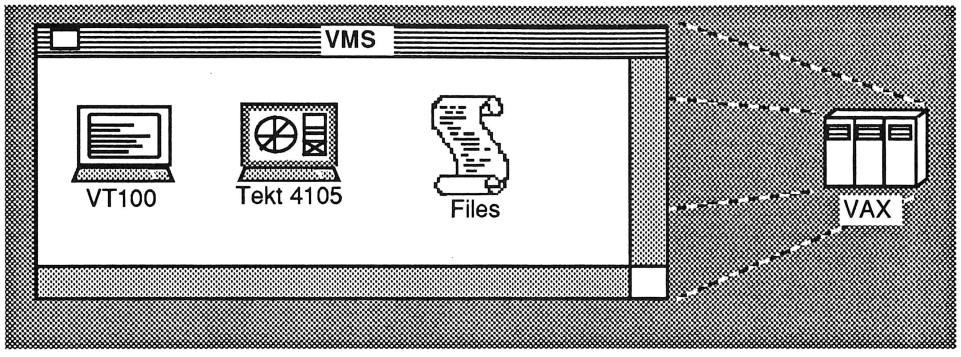
Whether to BinHex or not will be decided automatically based on whether the file is text only or a formatted data file. There is no way of deciding automatically whether a file transferred *to* the Macintosh is in BinHex format. The user will select the file after transfer and use a menu option to "unBinHex."

Transferring files from the Macintosh to the Cray will require the user to be signed on to both the VAX and the Cray.

Transferring files from the Macintosh to the Cray will require the user to be signed on to both the VAX and the Cray. When the Cray Finder is first loaded, it will prompt the user for his or her Cray and VAX sign-ons and passwords. This information will be used to log on and log off the machines as needed to accomplish the transfer. Perhaps a box can be drawn around the icons of the machines that the user is currently logged onto.

Terminal Emulation Modules

The next step is to add terminal emulation, perhaps concurrently with the file transfer modules to facilitate program debugging. The initial module will be a "plain vanilla" scrolling ASCII terminal. When a computer icon is opened (VAX, Cray, CFS), instead of a directory listing, the user will see two icons: one representing the terminal emulation and the other representing the directory listing.



Terminal emulators in the Cray Finder

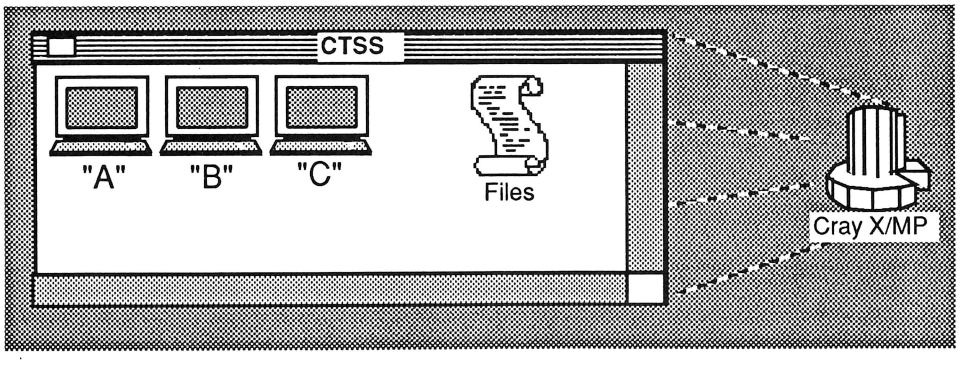
Double-clicking on the terminal icon will bring up the terminal emulator. Commands issued by the Cray Finder to the computers for log-on and file transfer will also be recorded in the terminal emulator, along with the system responses. This will be useful for debugging, and also for showing experienced users what the Cray Finder is doing.

The terminal emulator will have standard emulation features such as storage in memory of scrolled text, selection of text, "copy table," and so forth. VT100 emulation will be added to the emulator, as will Tektronix emulation (4010, 4105 color) somewhat later.

Users can run up to five sessions simultaneously.

The Cray terminal emulators will be similar to the VAX emulators, but with an added feature. CTSS lets users run up to five sessions (called A through E) simultaneously. Optionally, text from all five sessions can be intermixed and sent to the same terminal, where each line is prefixed with the session letter.

This means that the Cray Finder could have five terminal windows open, corresponding to the five session "prefixes." The text for each prefix would then be routed to the appropriate window. This, in effect, allows the Macintosh to handle multitasking, at least when accessing the Cray.



Multiple CTSS prefixes

Graphics Modules

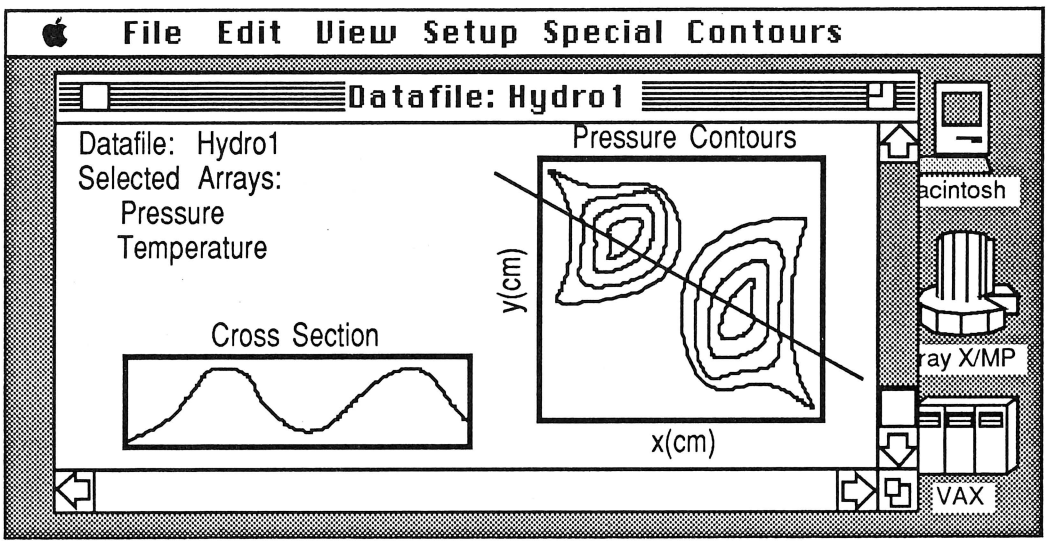
Understanding calculations by just reading numbers is impossible.

Graphic display of data is absolutely essential for supercomputer users. Crays produce so many numbers so quickly that understanding a calculation by reading printouts of numbers becomes virtually impossible.

Having interactive control over this graphic display is very important: it enables the supercomputer user to rapidly graph essential features of a calculation.

Although the Macintosh does not have the screen resolution or the color resolution of the higher-end workstations, it *does* have a highly interactive, friendly environment. Most of the higher-end functionality can be implemented on the Macintosh.

Contour plots on the Macintosh



There are currently three graphics modules proposed for the Cray Finder:

There will be three graphics modules in the Cray Finder.

Contour plots of raster files: Illinois has defined a standard format for raster files, which will be displayable on any NCSA workstation. Each pixel in the raster file may have many bits of intensity and/or color information. The Macintosh cannot display this file directly, but could display a contour map of the data. This program could, for example, let the user draw a line across the contour and display a cross section of the data at the line.

Conversion of raster files to MacPaint documents: For the special case of raster files with one bit per pixel, a program will allow the user to convert the file into a large MacPaint document. Dithering programs will be available on the mainframe computers to convert multiple-bit-per-pixel raster files to single-bit-per-pixel files that will be directly convertible to MacPaint files.

Interactive Graphics Post-Processor (IGPP): Recently, NCSA staff scientists obtained several mainframe-interactive graphics postprocessor programs. These programs enable supercomputer users to plot the results of their runs in any method they like, interactively.

We plan to have an interface for one of these mainframe programs running on the Macintosh. A much more ambitious project is to port sections of the interactive graphics postprocessor program onto the Macintosh. This would allow users to download selected parts of their data into the Macintosh and then "play" with it while off line. Only those parts of the program that the Macintosh can reasonably handle will be ported (no hidden surfaces, for example). The eventual idea is to merge the Macintosh program and the interface, so the user has no idea whether the postprocessing is being handled by the Macintosh or by a mainframe.

Programming Assistance Modules

How can NCSA make scientific programming on the mainframes more productive? One method is to make it possible for our users to compile and debug entirely on the Macintosh workstation, and download the program source code to the supercomputer when ready for production runs. A second method is to make Macintosh interfaces for mainframe programming tools such as compilers and debuggers.

Users may eventually compile and debug entirely on a Macintosh workstation.

Both methods should appear identical to the user: as with most other parts of the Cray Finder, the user should neither know nor care what processing the workstation is doing, and what processing the larger systems are doing.

The ideal programming environment includes at least the following features:

- Rapid programming cycle: from editing text to compilation to execution with a single click.
- Highlighting of code where the compiler detected errors.
- Highlighting of code that caused execution errors.
- Tracing of variables and program execution symbolically.
- Interactive syntax checking of code and subroutine calls.

Many workstation programming environments already have most of these features. Mainframe environments typically have much more complex programming cycles, and little or no linkage of the executable program to the source-code file.

Scientific Program Development on the Macintosh

For now and for the forseeable future, scientific programs on supercomputers will be written in FORTRAN 77. The Macintosh

The Macintosh can be used to test small code segments.

cannot realistically be expected to compile and debug a 10,000-line Cray FORTRAN program. However, it is clearly possible—and desirable—to do tests of small segments of code, or small test routines. The interactivity provided by the workstation environment is much better than what can be provided even with an interface to the mainframe, when working with small code segments.

Interfacing to Mainframe Programming Tools: After the scientific code has been coded and debugged on the workstation, the source is loaded into the supercomputer. We plan to provide Macintosh user interfaces for the mainframe compilers, debuggers, editors, linkers, and so on. The goal is to make this interface to the mainframe tools identical in appearance to the workstation programming environment described earlier. NCSA wants to make the transition between compiling and testing on the workstation and compiling and testing on the supercomputer instantaneous.

Supercomputer Applications Modules: We plan to have Macintosh interfaces for selected supercomputer applications. Those applications that are widely utilized but difficult to use will have highest priority. Applications that actually require mainframe power include data analysis programs, sophisticated graphics packages, and statistics programs. User productivity can be greatly increased by making these applications easier to use.

Mainframe applications that do not require mainframe power will be ported to the Macintosh to run as stand-alone Macintosh programs. Porting these programs to the workstations will reduce the load on the mainframe, making the supercomputer network that much more efficient and productive.

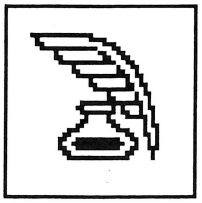
Interface to User Programs: Scientific users may wish to write Macintosh interfaces to their own supercomputer programs. NCSA plans to make this as easy as possible, by providing them with the needed programming environments and program libraries. Apple's MacWorkstation would probably be ideal for these users.

It should be possible to have this user interface run both under the Cray Finder and as a separate, stand-alone Macintosh application. We plan to support both approaches.

For further information, contact:

National Center for Supercomputing Applications

News and Notes



The **News and Notes** section contains information of general interest to those in the Macintosh community. In this issue, you will find an announcement of the Apple Programmer's and Developer's Association, Interactive Videodisc Training, and a variety of additional items. If you have information or know about events that you feel might be of interest to the general Macintosh community, please send your news along.

Apple Programmer's and Developer's Association

Whether you're an independent programmer or developer of Apple-compatible software or hardware, or just a person with an interest in programming and development, you need fast access to the latest technical information and development tools.

Now there's a place you can get it: the **Apple Programmer's and Developer's Association** (APDA). APDA is a one-stop source for the Apple programmer. Members have timely and easy access to many of the Apple development tools and much of the technical information necessary to speed development projects. No more wondering, "Where do I go for the information I need?"

Members have fast access to the latest technical information and development tools.

Whether you're looking for programming tools, utilities, technical notes, or manuals, now you have a single, convenient source. These Apple materials—along with selected products from other vendors—are available to all APDA members with one simple phone call.

Is APDA for You?

APDA was created to serve the needs of the entire Apple programming community. Membership is open to *anyone* who needs advanced tools and information in order to create Apple-compatible products.

The APDA was created to serve the needs of the entire Apple programming community.

APDA is ideal for:

- Apple enthusiasts writing software for commercial or noncommercial distribution.
- Educators and students developing curriculum-related materials.
- Consultants configuring specialized software or hardware for clients.
- Professional programmers and hardware developers.

Whether you make a full-time living from writing Apple software or just want to have ready access to the tools you need, membership in APDA could be just what you've been looking for.

Products Offered

Apple development tools, books, languages, and technical notes are available.

Here are some of the products you will be able to order as an APDA member:

- Apple II and Macintosh development tools, system software, beta versions of languages, technical notes, and manuals.
- Languages and development tools from a variety of third-party companies.
- Technical books, including the entire Apple Technical Library from Addison-Wesley.

Apple hardware and Apple-labeled software will continue to be available only from your local authorized Apple dealer.

Who We Are

APDA is a subsidiary of the A.P.P.L.E. Co-op, one of the first and largest Apple User Groups in the country. We have extensive experience with providing software and documentation to Apple enthusiasts, and will be working in full cooperation with Apple Computer to get you the up-to-date information and tools you need.

Benefits

As an APDA member, you will have fast, reliable access to the tools you need to develop Apple-compatible products.

There will be a quarterly newsletter.

A quarterly catalog/newsletter will give you full details on all available products. Regular bulletins will keep you up-to-date between newsletters. To order, use APDA's toll-free 800 number or our on-line ordering system. For your convenience, APDA accepts VISA and MasterCard. Better yet, you can subscribe to APDA's automatic ordering service and receive a copy of every new product or category of product as soon as it's in stock.

APDA's services will continue to expand to best meet the needs of our members.

Join Today

It's easy to join.

An APDA individual membership is easy to obtain, and dues are only $20 per year. Just photocopy and complete the application that follows, mail it to us with your check or credit card information, and start enjoying APDA's fast, easy, one-stop service for Apple developers.

If you need information, just give us a call at (206) 251-6548, or write to the address below:

Apple Programmer's and Developer's Association
290 SW 43rd Street
Renton, WA 98055

Apple Programmer's and Developer's Association Membership Agreement and Application

1. Purpose of the Association
The purpose of the Apple Programmer's and Developer's Association ("the association") is to promote the development of products that are compatible with Apple computers. Apple has authorized the distribution through the association of several technical documents and Apple software products (such as tools, utilities, and languages) that will be useful to developers of Apple-compatible products. These products are not available through Apple dealers. In addition, the association distributes Apple-compatible software products and solutions from third-party vendors. Thus, the association provides a one-stop service for programmers and developers working with Apple hardware.

2. Benefits of Membership
Membership in the association gives members the right to acquire Apple-compatible products and information services through the association. In addition, members will receive a quarterly newsletter containing notices of the availability of new products, detailed product descriptions, and a price list.

Agreement

By signing the Apple Programmer's and Developer's Association application, the applicant hereby applies for membership in the association and, if accepted by the association as a member, agrees to the following terms of membership:

Terms of membership

1. Membership Fee
Upon paying a yearly membership fee of $20, applicant will have the right for one year to acquire products and other services from the association at regularly published prices (less any applicable discounts), and to receive the quarterly newsletter.

2. Terms and Conditions for Apple Computer Products

a. Software products will be subject to a license agreement. Applicant acknowledges and agrees that all Apple Computer software products available through the association will be licensed—not sold—to the applicant, and that those software products will be shipped to the applicant with a license agreement attached. Applicant may obtain copies of the license agreement in advance from the association, if desired. Applicant agrees that by using the product, applicant will be bound by the terms of the license. If applicant does not agree to the license terms, applicant may return the unopened product to the association within three days of receipt, and receive a full refund.

b. Applicant's acknowledgment of the nature of the products distributed by the association. Applicant represents to the association that applicant is a software or hardware developer, and has substantial experience in computer programming and in using programmers' tools and utilities in the development of software. Applicant understands and acknowledges that the products distributed through the association are of a highly technical nature, and that neither the association, nor Apple Computer, Inc., nor authorized Apple dealers will offer the applicant technical support in the use of such products.

Applicant further acknowledges and understands that ***many of the products may contain errors,*** and that the documentation for the products may be nonexistent, incomplete (working documents only), or inaccurate.

Applicant understands that most or all of these products will be licensed or sold without warranty of any kind whatsoever, and that, pursuant to the license agreements under which the products are licensed, neither the association nor Apple Computer, Inc. will be responsible for any lost profits, lost data, or any other kind of direct or indirect damage that may result from the use of the products.

c. No right to distribute. Applicant acknowledges that under the license agreements, applicant will have no right to further distribute products produced by Apple Computer, Inc. and acquired through the association. Licenses for further distribution may be obtained from the Apple Computer Licensing Department, M/S 28B, 20525 Mariani Avenue, Cupertino, CA 95014.

3. Policies and Conditions for Third-Party Products
Software and other products from third-party producers range from full commercial versions to preliminary releases. License agreements, warranties, commercial distribution policies, and other conditions vary.

Name of applicant _____
(use company name if applicable)

Authorized representative _____
(if applicant is a company)

Address _____

_____

Telephone (_____)_____

Method of Payment

___ Check enclosed for $20 (made payable to APDA)

___ VISA # _____ Expiration date _____

___ MasterCard #_____ Expiration date _____

I agree to the terms and conditions of the Apple Programmer's and Developer's Association membership agreement.

Signature _____ Date _____

Return this signed application, together with your check or credit card information, to:

Apple Programmer's and Developer's Association
290 SW 43rd Street
Renton, WA 98055
(206) 251-6548

MacApp Developer's Association

The goal is to provide support for developers who are using object-oriented programming languages.

MacApp users are proud to announce the formation of the MacApp Developer's Association. Our goal is to provide technical support and information to our fellow developers who are using object-oriented programming languages. We will publish a regular (monthly?) newsletter addressing the concerns of MacApp users. We will provide, maintain, and support an object library for developers. As a developer's association, we feel it is important to provide an informal electronic-mail Q&A service for members, to supply answers to your MacApp questions. We will also provide support for the formation of local MacApp user groups.

Our initial organizing effort is being assisted by Apple Computer. But through your participation and support, we anticipate being able to function as a fully independent organization. Therefore, we are looking for volunteers able to contribute to the efforts described. If you have an article, an idea for an object for the library, or a tutorial to contribute to the newsletter, you should contact Eric Anderson (addresses are listed later in the article). If you wish to contribute an object to the object library, contact Kurt Schmucker concerning documentation details. If you would like to participate in the Q&A service, contact Greg Marriott or Scott Boyd. If you wish to form a local support group or are looking for local support, contact Carl Nelson.

Sign up for the first six newsletters.

To receive the first six newsletters, we are asking that you make a contribution of $15 to cover the costs of printing, production, and postage. Just send your name, address, and check to the MacApp Developer's Association. Details concerning access and distribution of the object library will be published in our first newsletter. Tentative plans call for the Q&A service to be conducted using MCI mail. (The cost of joining is low; there are no connect charges; and the price of sending a short, well-phrased question is a dollar or less.) To obtain an account, contact MCI at 800-MCI-MAIL. The MCI mail ID is MacApp Questions. In addition, the most useful Q&As will appear regularly in the newsletter.

For more information on MacApp, see the first article in this issue's **Technical** *section.*

MacApp Developer's Association
P.O. Box 23
Everett, WA 98206-0023

MCI: Carl Nelson
USENET: uw-beaver!tikal!sigma!chn

Kurt Schmucker
PPI
2516 Vineyard Lane
Crofton, MD 21114

BIX: kschmucker

Greg Marriott/Scott Boyd
3420 D Sandra
Bryant, TX 77801

MCI: MacApp Questions

Eric Anderson
P.O. Box 1145
Montpelier, VT 05602

MCI: Tomevs

BITNET: GBMARRIO@TAMVENUS.BITNET
boyd@tamlsr.bitnet
CSNET: boyd%tamlsr@tamu.csnet
General: boyd%tamlsr%bitnet@wiscvm.wisc.EDU

Poor Tom: An Associational Data Base

Overview

Poor Tom gives its data items a "position in space."

Poor Tom is a public domain program that tries to demonstrate a fundamentally different method of storing data. Whereas a normal data base can be thought of as a set of predefined slots (fields) within a series of boxes (records), Poor Tom gives each of its data items a "position in space." A normal data base relates its data by sequencing them in linear orders or with solid links; Poor Tom relates them by "distance."

In Poor Tom, each record has coordinates that fix it in eight-dimensional space. The distance between two records corresponds to the formula used for points in two or three dimensions. Relationships can be changed either by directly changing the coordinates of either word, or by ignoring one or more of the dimensions.

Records are related by association.

Think of a simple game of word association. What makes you relate one word to another? If you make the gross simplification of thinking of words and ideas as floating freely within your brain, with associated words close together, dissimilar ones far apart, then Poor Tom could model such a system.

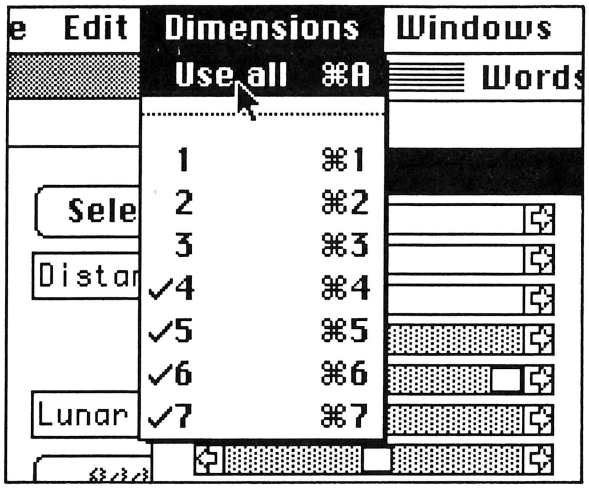
The program was intended for use with words, but could conceivably be used for other types of data. For best readability, the Monaco 12-point font should be in your System file.

Menu Items

File currently only allows exiting from the program.

Edit allows for the cutting and pasting of words within the program and between applications.

The Dimensions menu



Dimensions allows you to choose which coordinates Poor Tom will use in its operations. You can use from one to seven coordinates in such activities as searching around a word for its relatives and adding related words. The eighth dimension includes the word type, and can be activated separately from each window.

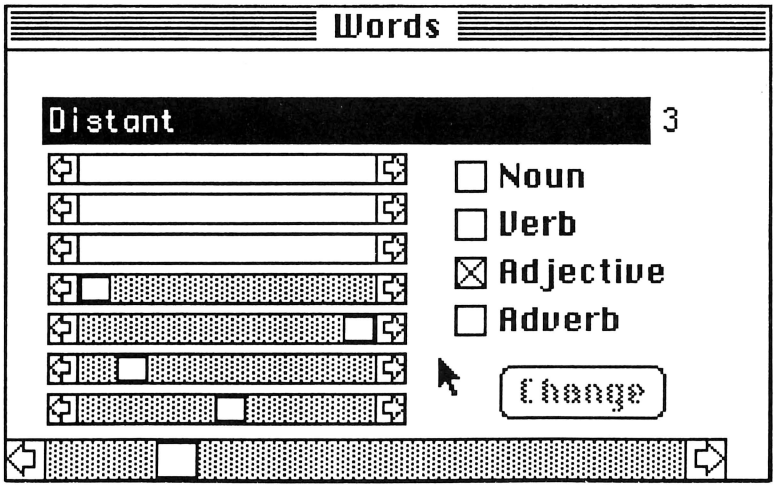
Windows allows you to change windows from the menu, since the small ones can get hidden behind the Learn window.

The menu is a program resource. You can add your own terms.

The menu is a program resource, so you may change any text. (I can especially see giving each dimension its own special term, such as "color," "abstraction," and so forth.)

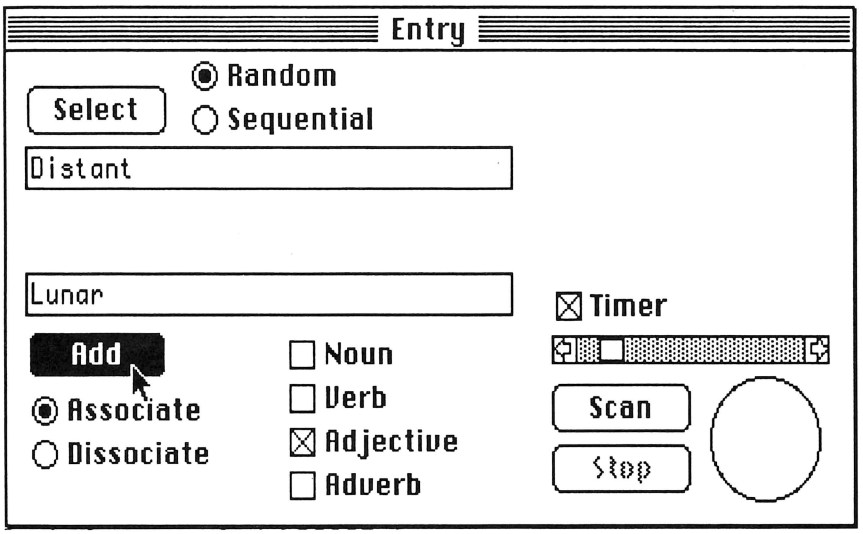
Words allows you to view your data in a relatively orderly manner. The lowest horizontal scroll bar takes you through the records. The record number appears in the upper right. You can change the word type (Noun, Verb, and so on), and a word can be of more than one type. You may change the text also, but must confirm such modifications by pressing the Change button. You can see each of

The Learn window



the position coordinates, but you may not change them through this window. Altering the dimensions used (from the menu) immediately affects the dimension display in this window as well as the graph display in the **Learn** window. Words may not be deleted.

The **Entry** window is the only place where you may add new records. You enter your words into the middle Add rectangle, and select the word type with the check boxes beside it. You must relate each new word to an existing one by choosing to place your new word close to (associate) or far from (dissociate) the word shown in the upper rectangle (the "Entry Root" word). Pressing the Add button places the word at the end of the data file. The chosen relationship affects only the currently active dimensions—all other dimensions are randomly set.



The Entry window

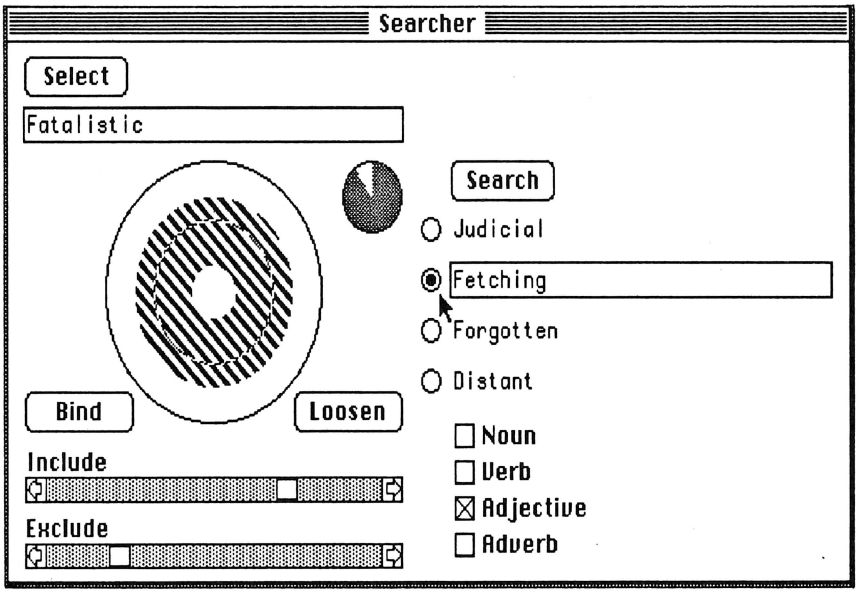
You may change the Entry Root word by pressing the Select button, going through the data base randomly or sequentially, depending on the active radio button. You can simulate a word association game by activating the Timer feature and pressing the Scan button. The window will show a word from the data base after each selected time interval. The timer pauses whenever you begin to enter a new word.

The **Learn** window works as both your tool to "teach" the data-base relationships and the port through which you discover relationships. As in the **Entry** window, the word in the upper left rectangle functions as the root of activity. It is the base around which other words are measured. The four word areas in the middle left portion of the window are where the related words appear. The graph, located middle right, shows distance relationships between root and other words.

To scan for word relationships, you must first choose a root, either randomly by pressing the Select button, or in a more deliberate

manner by selecting in either the **Words** or **Entry** window. (There is one other method for choosing a root, which I will describe shortly.) You must select an area in which to look for related words: you use the Include and Exclude scroll bars to do this. The graph shows the area to be covered; the center of the circle represents the root. Note that the pattern changes depending on the number of dimensions you are considering. Also realize that distances are normalized to take into account the number of active dimensions, so that the edge of the circle will always represent the extreme distance from the root. Pressing the Search button starts a scan through the entire data base for all words within the given range. The small circle to the right of the graph shows the percentage of the entire data base scanned as the found words, if any, appear.

Searching the data base



If you find any words within the designated distance from the root, you can change their relationships to the root by pressing a radio button and either the Bind (bringing the word closer to the root) or Loosen (just the opposite) buttons at the bottom of the graph. You can see the change represented on the graph as an expanding/contracting circle. Binding and loosening affects only the found word, not the root, and only in the dimensions you have activated. You can make any of the four found words the root by pressing its radio button and then pressing the Select button.

You can narrow the types of words selected by using the check boxes at the lower right of the window. You may select more than one type. All types will be used should you not choose any word type.

File Structure

Poor Tom uses a simple text file for its data. Currently, the file must be called "Words Data." Each record is 40 bytes long, consisting of eight 1-byte "dimensions" and a 32-character "word." The first two records are reserved for future use. The first seven dimensions can each contain a value between 0 and 255; the numbers loop such that moving a coordinate up past 255 will bring it through 0 and "opposite" coordinates will be 128 apart. The eighth dimension holds the word type in bits 0 to 3.

To create a new data file, rename the "Empty Words" file as "Words Data," or create a 120-byte empty text file called "Words Data." The first word will be blank—which you may, of course, change.

Warning

You are hereby warned that Poor Tom is more "provocative" than "practical." I have even found it fairly useless as an inspirational/creative aid, the purpose for which I first envisioned it. It can be quite frustrating for even the simplest "mental models." (I think in large part because of my relative inexperience with anything but "straight-edged" data.) You will find, if you spend any time experimenting with it, that you've got to play it a lot looser than with other computer programs.

Poor Tom is more provocative than practical.

I'm learning that the basis behind a good associational data base isn't in the quantity or even the quality of the words, but in playing on the **Learn** screen for a while with your data, getting the words related in the right way (for "right," read "instinctive"—playing fast seems to make it better). Because each time you change the relationship between a word and a root, you also alter its relationship to all other words, good learning must be a slow, careful process.

All this said, I hope the program does inspire others to explore what I regard to be a fascinating subject: alternate data-base styles.

Etcetera

The MacForth source code is available to anyone who wants it. I would also like any suggestions and comments about the program.

The MacForth source code is available to anyone who wants it.

Future versions will include use of the standard file interface (regular Open menu item) and option to load words from another text file.

For further information, contact:

Glen Harris
718 Lexington Ave.
El Cerrito, CA 94530

The University of Utah Department of Music

The music department is on the forefront of electronic music development.

The University of Utah Department of Music has been involved with electronic music since 1970 when Dr. Vladimir Ussachevsky joined the faculty as composer in residence. He left his position as director of the Columbia-Princeton Electronic Music Center and arrived at the University of Utah with about a dozen boxes of electronic sound-processing equipment. Under his direction, the music department has remained at the forefront of electronic music development. Some of that original equipment is still being utilized in the composition of electronic music; however, the foundation of our program is now built around digital synthesis equipment and computers such as the Macintosh.

Two years ago, Professor Ardean Watts set out to investigate the possibility of integrating digital synthesis technology (then used primarily in the pop music industry) into a classical music curriculum. It was his intent to use the relatively untapped potential of computing devices in the development and control of orchestral sounds both in the classroom and in performance.

The most recent result of the department's commitment to this new technology was an interdepartmental production of Carl Orff's "Carmina Burana." As a combined effort of both the music department and the Utah Ballet, this production was a world premiere of new choreography and utilized a MIDI-equipped Macintosh, connected to a variety of synthesis equipment, as the orchestra. Similar setups have been and are used in recording studios and even in some performances; however, if the chorus, the dancers, and the conductor all had to follow the computer's beat-perfect playback of the score, all artistic spontaneity would be lost. In order to give control of the performance back to the conductor, a MIDI timing device (the Milton Box) was built by Milton Watts (Professor Watts' son). Taking his cues from the conductor, Professor Watts tapped out each beat and the Macintosh responded by controlling eight synthesizers and a drum machine, one beat at a time. The Macintosh is used as a multitrack recorder.

Using state-of-the-art technology in untried situations can be nerve-racking.

As can be expected when using state-of-the-art technology in untried applications, Murphy's Law was in effect. The final working version of the Milton Box was completed just one hour before the curtain went up on the first performance. Still, that first performance came together like magic and was a resounding success, as were the next three. The audience was thrilled and awed that a little Macintosh could help produce such beautiful orchestral music.

The University of Utah Department of Music is working toward computer literacy for music students through continued development of specially designed labs. Professor Watts, Music Department Chairman Dr. Ed Thompson (a musician and physicist), and Alan

Done (the staff musical engineer), are implementing two orchestration and composition studios. Each has a full complement of digital and analog synthesis equipment under the control of the most sophisticated MIDI software available running on Macintosh and IBM computers.

A student lab of Macintosh computers is under construction.

A student access lab (currently under construction) contains an additional six Macintosh computers in individual carrels, each interfaced to a Yamaha digital MIDI synthesizer and running the same high-quality software. These Macintosh computers are all networked to an Apple Hard Disk 20 and an ImageWriter II printer through a network that uses MacServe. There are currently three classes offered by the department providing instruction in various computer music applications. In addition, many of the freshman and sophomore music core courses will require students to complete assignments in the student labs.

Future concert plans include using the synthesizer.

The music department has also integrated the Macintosh into other administrative and departmental functions. For instance, future concert plans include using the synthesizer to accompany the a cappella choir in both concerts and recordings. Classes that will use the Macintosh labs include Music Composition, Introduction to the Synthesizers, Introduction to Recording Techniques, and Sound Composition. Administrative uses of the Macintosh include department budgets, class schedules, and word processing. All concert programs and graphics for departmental use are produced using commercial software and the LaserWriter printer. A multiuser data base recording the sheet music and record library is being set up to allow simultaneous accessing and updating of information.

Alan Done, the department's musical engineer, is in the prealpha planning stage of writing computer-aided instruction and composition programs using several programming languages. The music department is coordinating efforts with another Apple University Consortium university to produce CAI and courseware using an authoring system to be written in Pascal.

For further information, contact:

Alan Done
Music Department
204 Gardner Hall
University of Utah Campus
Salt Lake City, UT 84112
(801) 581-6762

Top of the Morning to You, Macintosh

Trinity College (the University of Dublin) has joined the consortium.

Apple Ireland is pleased to announce that Trinity College (The University of Dublin) has become a member of the Apple University Consortium.

Better known as Trinity College, Dublin, the university was founded in 1592 and is the oldest university in Ireland and one of the oldest in Western Europe. It stands in the heart of Dublin and forms a compact academic community, at the same time an intimate part of the city's life.

Some of the most famous men in Irish history were educated in Trinity: patriots such as Wolfe Tone, Thomas Davis, and Douglas Hyde; world-famous writers such as Dean Swift, Oliver Goldsmith, and Samuel Beckett; and scientists such as Rowan Hamilton, the mathematical scholar, and E.T.S. Walton, who won the Nobel Prize for his work on the atom.

Trinity students, faculty, and staff are fully aware of the spirit of the consortium and wish to be full participants in all AUC projects, so please include them in all mailings and keep them informed of any activities you may be aware of. They are particularly interested in using Macintosh as a teaching tool.

For further information, contact:

Michael Doherty
Computer Lab
Trinity College
College Green
Dublin 2
Republic of Ireland
Ph. 01-772941

Apple Ireland seeks aid.

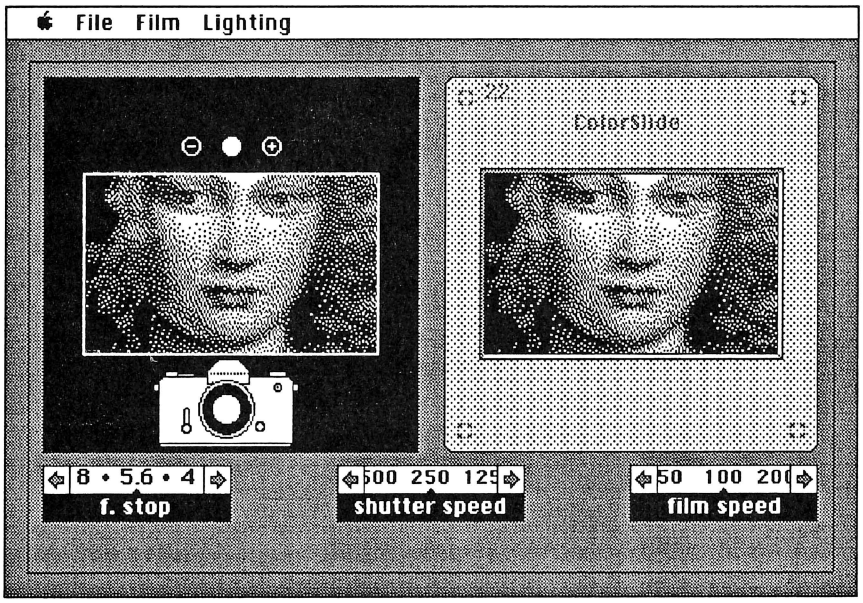
In addition, if any of you have any suggestions about what we (Apple Ireland) can do to help them participate fully, please contact us at AppleLink: DUBLIN.

Macintosh Simulated Camera

By Craig Hickman
University of Oregon
Department of Fine and Applied Arts

The Macintosh computer provides opportunities for sophisticated simulations.

The past decade has seen an explosion in computer software and hardware designed to teach everything from algebra to car mechanics. The techniques used, however, rarely tax the capabilities of computers, and are for the most part firmly rooted in teaching methods developed before the computer, such as drill and practice. But now that many microcomputers are equipped with sophisticated graphics capabilities, it becomes possible to explore one of the most exciting and effective learning techniques: simulation.



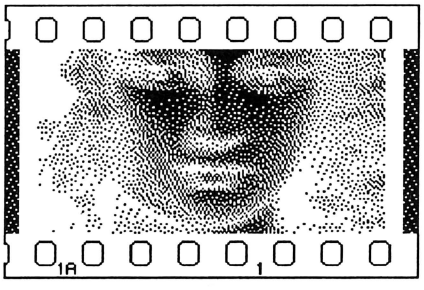
When most people think of computer simulations on microcomputers, they think of electronic games. Indeed, most computer games are simulations—Pong of ping pong, and Missile Command of World War III—but they involve little educational activity. There are some educational games, but their goal is mainly to "entertain"—to teach something while students are not looking. Often the game has little to do with the subject being taught, and very few games attempt truly to simulate the target activity.

While fast action and competing to get a higher score account for the thrill of playing normal computer games, the thrill of a true simulation comes from feeling that one did the "real" thing. One of the best-known educational computer simulations is the popular

Microsoft Flight Simulator, which reproduces the activity of flying an airplane. Unlike a traditional computer game that only attempts to mimic a few superficial aspects of an activity, Flight Simulator attempts to mimic the activity's more subtle aspects. The Flight Simulator screen displays all normal aircraft instruments, which respond in a realistic way: the operator takes off and lands at "real" airports; the flight takes place in "real time" (for example, it takes six hours to fly from New York to Portland).

Camera helps students understand the proper way to use their camera.

My interest in computer simulation recently led me to write a program relating to one of my own teaching areas, photography. One of the hardest things for beginning photography students to learn is the interrelationship of f-stop, shutter speed, film speed, negative density, and exposure. The terminology is confusing (for example, when you "stop down" you set the aperture to a larger number), and by the time students see their results, so much time has passed that they have forgotten what they did. So setting up a camera seemed like an ideal activity to simulate by computer.



On one side of the screen, I established an area that represents the camera viewfinder with a light meter, and on the other side an area representing a 35mm slide. Below are the f-stop, shutter speed, and film speed controls. The control settings may be changed by placing the cursor on them and clicking with the mouse. As the settings are changed, a digitized picture inside the slide mount becomes lighter or darker, representing the picture resulting from that exposure. By playing with the controls for a while, the student grasps the relationship between the controls and exposure in a practical way.

There are two additional practice features in the program. Since students often get confused about which f-stops let in more light and which let in less, I also created a small representation of a camera with an aperture that changes size as the user changes the f-stop. The user may also elect to view the image in negative rather than as a positive slide, in order to study the way in which exposure affects negative density.

Camera is in the public domain.

I developed Camera as an example program for my class, Programming for the Arts, whose goal is to create software to assist the artist. Camera is in the public domain, and anyone may secure a copy of it by sending a blank Macintosh disk, along with return postage.

For further information, contact:

Craig Hickman
Department of Fine and Applied Arts
University of Oregon
Eugene, OR 97403

Desktop Training: Interactive Authoring

The Macintosh has been the catalyst of some of the most innovative and creative educational programs ever created for a microcomputer. Capitalizing on the powers of the Macintosh, Edudisc has developed an authoring system that incorporates the computer's visual interface and ease-of-use features. Desktop training is the next frontier for educators and trainers. Desktop training allows them to apply their instructional-design talents without being hampered by complex authoring languages. It fills a need to provide creative, customized, computer-assisted presentations, reasonably priced and easily developed. It was designed for instructors and courseware developers at all levels of education and business. Instructors can incorporate text and graphics from other Macintosh programs and courses developed with this system.

The user can control a variety of videodisc players.

Edudisc's authoring system allows the user to control a variety of videodisc players, and is integrated with a full-featured authoring program. It's designed to provide an inexpensive way for educators and trainers to develop and produce their own interactive video courseware, utilizing their existing video and Macintosh resources. Compared with other such systems, Edudisc products offer a greatly improved learning curve and a significant decrease in development time.

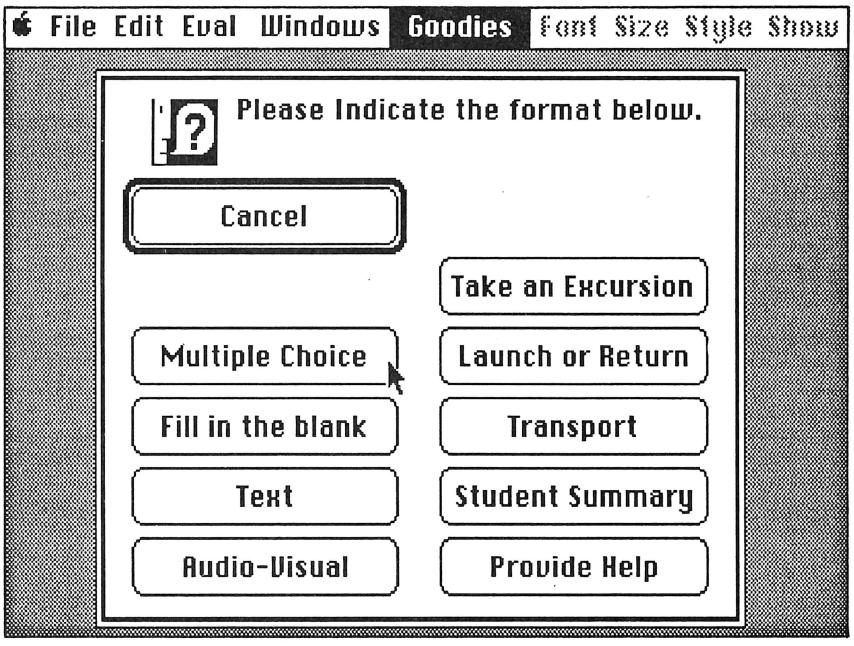
An authoring system allows nonprogrammers to use a computer for creating and delivering instruction.

The software package, **Mentor/MacVideo**, works to form an authoring system, rather than an authoring language. As defined by Stephen Floyd *(The Videodisc Monitor,* March 1986), "an authoring system allows nonprogrammers to use a computer for creating and delivering instruction and presentations. In simplest terms, an authoring system automatically translates the author's input into code that can be used by the computer." Mentor does this by using icons, menus, windows, mouse/keyboard input, and a single cable videodisc player interface.

Mentor Features

Upon opening Mentor, the courseware author will find the menu structure similar to that of standard Macintosh applications. Selecting New from the File menu starts a new course, and from this point on, Mentor is similarly straightforward. It lets users follow a linear or relational format, or use a combination of the two. Instructional formats include multiple choice, fill in the blank, text, graphics, video, and "relational thinking," which allows a dynamic approach to a topic. After naming the course and a topic, selecting Add New Item from the Goodies menu allows the addition of formats from the menu pictured on the next page.

The Goodies window



Here is a more detailed explanation of the formats and special branching features:

Enter text.

Text windows allow text input directly from Macintosh word-processing files, including the variety of fonts, sizes, and styles available on the Macintosh. (You can also create your own fonts.) Information put directly into Mentor windows may also be saved as text files for later use. Text windows may be scrolled, sized, and placed anywhere on the screen. Texts may also be "spoken"—using the SmoothTalker or MacinTalk speech synthesizers—or overlayed on video.

Use multiple choice questions.

The **Multiple Choice** format comes in two sizes: full or half screen. It accepts up to six answers for each question and multiple correct answers. The answer choices may be randomized each time the question is shown. Feedback branching is easily designated and may be based on a single question or a series.

Fill in the blanks.

Fill in the Blank allows for multiple-word answers and accepts up to several thousand phrases as answers. Feedback branching is easily designated and may also be based on a single question or a series.

Branch elsewhere if you wish.

The **Take an Excursion** format allows for up to 12 separate branches from an excursion window. It's useful for ready-reference information where excursions may be nested within each other for hierarchical classifications. A single, large ready-reference course can store several thousand items of information, each one accessible with four mouse clicks. This format may also be adapted for simulations in which the users' responses can take them on various pathways.

The **Audio-Visual** format is where videodisc clip names are entered. This format also holds digitized pictures, art, or other graphics.

These formats may be automatically sized, placed, and timed according to the author's desire to control the flow and presentation of the information while in viewing mode.

The **Transport** feature allows for conditional branching to any point in a course, or branching from multiple points to a single location. It is used for branching users back and forth to information throughout the course, based on the author's criteria or user selection.

Link courses with your own programs.

Launch or Return is a feature that links Mentor courses to other Macintosh applications or to your own programs that run on a Macintosh. You can set up a launch during a Mentor course to automatically go to another application and return to Mentor at the point where the viewer left off. It's also very effective for linking courseware modules created with Mentor in which course evaluation information may result in up to 12 possible branches after the return.

Get viewer feedback.

Student Summary is for feedback after users have completed a course. Summaries are placed in desired locations and appear at the end of the course to suggest remediation or reinforcement for user actions while in the course.

Custom menus.

Provide Help allows authors to create custom menu items that appear on the Help menu at designated times. They may contain reference information, including text, graphics, or video, during the lesson.

A complete **evaluation** function is provided for linear lessons, keeping track of the total time spent on a course, the percentage of correct and incorrect attempts, numbers of correct and incorrect attempts, and the total score for all users.

Edit in real time.

Editing can be done in a real-time mode, which allows you to run a course as the viewer will see it and edit as you go. It is not necessary to return to an off-line mode to make changes to any information in the course. **Launch Another Program** allows switching to another application, such as MacPaint, MacWrite, or MacVideo during the editing process. For example, you can go directly from Mentor to MacPaint, copy a picture, return to Mentor and paste that picture into the course without having to return to the Finder. Editing of video "clips" can also be done through this simple launch procedure, which can take the author to MacVideo to make the necessary changes and automatically back to the editing point in the course.

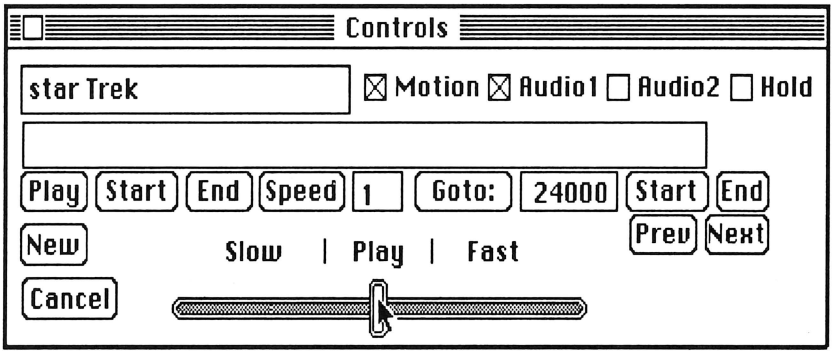
Mentor is complex, though not difficult to learn. It offers virtually limitless options and creative possibilities to authors.

MacVideo Features

The mouse can be used to control the video.

The second part of the Mentor package allows control of the videodisc functions from the computer. The mouse can be used to drag a speed throttle and control the direction of play. This control and the radio buttons allow for easy editing of still frames or full motion video/audio "clips." Frame numbers may also be entered from the keyboard for searching directly. MacVideo automatically keeps track of frame numbers for each clip. The frame numbers are stored as the beginnings and ends of clips and may be printed out as a playlist.

The video control panel



The ability to create video "clips" allows the user to select sections of an existing videodisc and use them as a resource in a Mentor course. Clips are titled with common words and stored in a file. Clips can be joined together into "Groups" to allow easier editing, and to allow sequences of stills and/or motion to be played back as one element. This adds enormous instructional impact to the Mentor program, allowing the integration of still and full-motion video, as well as two audio tracks. Existing videodiscs can be rearranged and, in effect, reedited.

MacVideo gives the Macintosh control over the Sony LDP-1000, 1000A, 2000 series and Pioneer LD-V6000, 6010 videodisc players (interface with the new LD-V4200 will be available in January 1987). It also controls the Panasonic TQ-2023F, TQ2026F videodisc recorders, and TQ-2024F, TQ2027F players.

This package allows "repurposing," or reworking of generic videodiscs to create customized lessons. The text and graphics information is easily updated on the Macintosh, and randomly accessed video segments are captured from the videodisc and displayed on a separate monitor.

Edudisc's recording system incorporates the Optical Memory Disc Recorder's (OMDR) instant disc-recording capabilities to significantly shorten application development time, and cut costs dramatically. Edudisc provides a user-friendly interface for

controlling all aspects of recording and playing. Users can record or play back any sequence of still frames or moving video at the click of a mouse.

Disc recording services are available.

Optical disc recording and **replication** services are available for customers who do not want to purchase their own Panasonic disc recorder. Edudisc records on single-sided, 8-inch optical videodiscs that are playable only on the Panasonic TQ2023F, TQ2026F (OMDR), and TQ2024F, TQ2027F (OMDP). Disc capacity is 24,000 still frames or 13.3 minutes of motion and audio. Original program material can include film, videotape (1/2, 3/4, or 1 inch), microfilm, 35mm slides, flat art, photographs, computer graphics, and animated sequences. Simply, you can record anything with a video camera or from videotape.

Courseware Delivery

A run-time Mentor comes with the full package for presentation of courses. Viewers receive a disk with the course and simply click on the course icon to run the lesson. All they have to do is point to check boxes and click, or key in answers to fill in the blanks. There are no commands for viewers to remember; therefore it requires only the most rudimentary knowledge of how to operate the Macintosh—pointing and clicking—to view a lesson.

Advantages

The Edudisc package provides a complete authoring system.

Benefits of the authoring system:

- Combines a powerful, yet user-friendly microcomputer with a single-cable/videodisc interface.
- Eliminates the need to modify the Macintosh with any hardware, because it's not a dedicated system.
- Enables novice users to quickly become accustomed to developing branching formats and interactive lessons, through authoring.
- Allows "on-the-spot" editing for both course content and video clips.
- Enables the creation of complex levels of interaction, through multilevel branching.
- Allows enhancement and greater utility of courseware, through integration of many other Macintosh programs. (It allows authors to add commercial applications or programs they have developed.)
- Enables networking of courses on a central file server.
- Offers a cut-and-paste way to create lessons.

- Provides upward compatibility through software upgrades; protects your investment in time and effort.

Demonstration Program

The demo version of Mentor/MacVideo has *fully* activated features. All of the authoring steps may be utilized and a complete course created and run. The only limitation is that once the course file is closed, it may not be reopened for editing or additional use without Full Mentor.

Applications

Edudisc Desktop Training is being used in universities, medical schools, technical schools, and business and government training departments throughout North America, and in some European sites.

Courseware topics include diverse medical subjects (pathology, pediatrics, dentistry, immunology), engineering, languages, English composition, accounting practices, administration, correctional officer training, and Macintosh training.

Related Products

Three additional products round out Edudisc Desktop Training:

MacAuthor is a stand-alone version of the Mentor authoring software that includes all of the same formats and features, but without a videodisc interface.

A still-image data base.

Portfolio is a still-image data base for cataloging still frames. It provides fast access to images via keyword searches. The Search menu allows for single or combination word searches, and standard editing after a file of images has been created. Catalog up to 24,000 stills per 8-inch disc. Create your own videodisc directory on the Macintosh.

A time-lapse video program.

TimeMaster is a time-lapse video program to record laboratory experiments, historical events, or any visual sequence directly onto the Panasonic recordable disc. It's applicable to any type of video time-lapse work and eliminates costly tape editing. These professional-quality sequences may also be used in courseware development with Mentor/MacVideo.

System Requirements

- Macintosh 512K or Macintosh Plus with external drive.
- Videodisc: Panasonic TQ2023F, 2026F; TQ2024F, TQ2027F Pioneer LD-V6000, 6010 [LD-V4200 available January 1987] Sony LDP-1000(A), 2000 series; interface; NTSC video monitor
- Software: Mentor/MacVideo $595 (site licenses available for multiple users)

Future releases will include a flowcharting font, graphics "hot spots," single-screen video integration and more curriculum formats, such as matching, and compare and contrast.

MacAuthor and Mentor/MacVideo Q&A

What equipment do I need to get started?

If you already have a Macintosh (512K or Macintosh Plus), you're set for MacAuthor.

Hardware needs.

If you want to create interactive video courseware, then you'll need a videodisc player, monitor, cables, RS-232 interface, and Mentor/MacVideo software. The Full MacAuthor and Full Mentor programs enable you to create and edit courses. Then you can present your courses in playback mode with the "Viewer" MacAuthor or Mentor. We supply the authoring system, documentation, and sample courses.

What if I already have a videodisc player and want to buy only the software?

Software needs.

The single-copy fee is $595 for the combined authoring and viewer-playback version. Check that your videodisc player is compatible with our software. You should also call us if you are looking for a particular topic to create an interactive video course. There are many disc titles available that provide excellent source materials to repurpose for your particular needs.

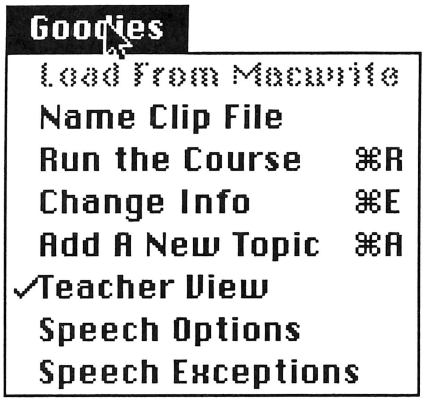
What about creating a course on one Macintosh and presenting it?

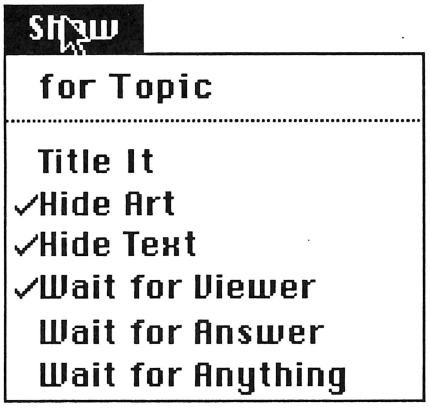
Additional run-time copies of MacAuthor ($20) and Mentor ($50) may be purchased for viewer delivery of your courses.

What extra benefits do I get with a site license?

Site licenses.

- Unlimited use of the software throughout your department. Lack of copy protection permits usage on a network in multiuser environments.
- No charge for updates and enhancements, unlike with a single copy.
- One day of training, and technical and consulting support are included in the site fee. Includes consulting on your videodisc production, duplication, and courseware development. Personal service is given at all times.
- Quarterly newsletter about our network of users and items about the software and new applications.
- Unlimited telephone support.





What if I want to sell my courseware outside of my institution?

With a site license, you may distribute your courseware outside of the institution and pay Edudisc a royalty payment. We will assist you in marketing and distributing your completed courses.

What advantages are there in having my own disc recorder?

It reduces your production costs by allowing you to record directly from any video source—slides or pictures—rather than having to transfer images to 3/4- or 1-inch tape.

It eliminates complicated mastering procedures such as time coding, control tracking, tape lead-in/out, audio modulation, and so forth.

It allows absolute control over the recording process.

It offers you the flexibility of recording when and where you want.

It facilitates updating of discs with new information on an ongoing basis.

It enables easy transfer of recorded material from one optical disc to another.

It helps you to achieve high-quality images with less effort.

Will the software assist me in videodisc recording?

Yes. MacVideo knows when your Macintosh is connected to a Panasonic videodisc recorder. There are special menus and dialogs that are enabled to allow you to record easily from your video source. Recording becomes fully Macintosh controlled.

Will Edudisc record a disc for me?

Yes. If you do not want to purchase your own videodisc recorder, send us your video, and we will record it onto an 8-inch optical disc. All you need then is a Panasonic player. The basic fee for this service is $500. It's not necessary to have a fully edited tape(s). Edudisc will record video segments from different tapes, as long as their combined time is no longer than 13.3 minutes. (Note that Panasonic recordable optical discs do not play on LaserVision videodisc players.)

For further information, contact:

Edudisc, Inc.
3501 Amanda
Nashville, TN 37215
(615) 269-9508

Edudisc Authoring Work in Progress

Blood pressure diagnostics

Carlene Grim, of Shared Care Productions (25 Coveview Drive, Rancho Palos Verdes, California 90274; (213) 544-3207), is transferring some of her popular videotape materials to videodisc using the Edudisc system. Her area of expertise is blood pressure procedures and diagnostics. She plans to develop other interactive video medical topics.

Academy trainees

The California Department of Youth Authority has implemented the first phase of a four-year program to design interactive video courseware for Basic Academy trainees. Director Si Mariano is establishing a totally integrated classroom and office operation, with 50 Macintosh systems at the training center in 1987–88. The Department of Youth Authority has a Mentor/MacVideo site license provided through an interagency agreement with San Joaquin Delta Community College in Stockton, California.

Safety procedures

The Tennessee Valley Authority's technical support branch is using Portfolio to record and catalog still images. The first project will be to catalog 20,000 images of the group's vast slide and videotape library on everything from crane safety to water recreation. Edudisc is providing the recording training and services.

Remedial grammar

The University of Texas at Arlington has a MacAuthor site-licensing agreement for its English lab. Faculty there are developing remedial grammar exercises and writing composition lessons.

English as a second language

Brown University's language lab has produced its first Mentor course on TESL (Teaching English as a Second Language) using a CAV LaserVision disc titled "Citizen Kane."

Art applications

The Memphis College of Art purchased a videodisc recorder to provide mastering services for its consortium of art schools.

Video instruction

Uniform Services University in Bethesda, Maryland, has purchased a site license and a Panasonic recorder, and is about to embark on an ambitious program of video-oriented instruction.

Work Not Yet in Progress

With help, our ideas can become great.

Sometimes we lack the necessary expertise; sometimes we want the opinions of others before we start; sometimes our "good idea" can be made a "great idea" by a simple addition from another person.

This section contains information about ideas residing somewhere in our gray matter that we would like to see further developed. It will appear each issue. Consider it a good example of a network—a "people" network, not a computer network.

What a great resource—people willing to work together. Participation is voluntary, so it can only work if you help. Send in your information now (the address is in the **Editorial**). Do it today. Don't put it off!

Video discs

Expert systems

Biofeedback

Electronic games

Name:	Peter Olivieri
Address:	Boston College Computer Sciences F430 Chestnut Hill, MA 02167
Telephone:	(617) 552-3907
Electronic Address:	BITNET: OLIVIERI@BCVAX3 APPLELINK: WHEELS
Affiliation:	Boston College
Areas of Expertise:	Teaching Programming (a variety of languages) Quantitative analysis/MIS/ Computer science Videotape production Writing
Ideas/Interests:	A videodisc-to-Macintosh connection Expert systems in sports Biofeedback Multiperson electronic games

Name:	Eric T. Lane	***Holograms***
Address:	Physics Department University of Tennesssee at Chattanooga Chattanooga, TN 37403–2598	***Animation in science*** ***Fourier transforms***
Telephone:	(615) 755-4523	***Chemical chromatography***
Affiliation:	The Center for Excellence for Computer Applications	
Areas of Expertise:	Animation graphics (Apple II, Macintosh, Rainbow) Science education (teaching, tutorial development) Physics (teaching, consulting, tutorials) Programming (C, BASIC, Pascal, FORTRAN, Assembler) General systems analysis (theory, applications)	
Ideas/Interests:	Fourier transform laboratory Holograms on the Macintosh Relativistic electromagnetic waves Chemical chromatography simulation Animation in science education tutorials	

Name:	Nello Lucchesi	***DEC VAX***
Address:	Household Commercial Financial Services Household International 2700 Sanders Road Prospect Heights, IL 60070	***DSS applications*** ***Financial analysis*** ***Sales-force management***
Telephone:	(312) 564-5000	
Affiliation:	Senior vice president, operations	
Areas of Expertise:	Financial analysis Accounting Decision support systems (DSS)	
Ideas/Interests:	Using Macintosh Plus systems as terminals with DEC VAX DSS applications and systems in the areas of sales-force management, financial analysis (data base, graphics, statistics in an object-oriented computer language)	

Videodiscs and music

Expert systems in music

Music analysis

Sailing

Name: Kenneth Rumery

Address: Northern Arizona University
Music Department Box 6040
Flagstaff, AZ 86011
or
9 Pine Ridge Drive
Flagstaff, AZ 86001

Telephone: (602) 774-1584 (home)

Affiliation: Faculty, Northern Arizona University

Areas of Expertise: Teaching
Writing
Music composition, theory, analysis
Music CAI
Electronic/Digital music
Conducting
Sailing

Ideas/Interests: Videodiscs for music instruction
Expert systems in music
Music processors
Music analysis simulation
Sailing simulations

Keyboard remapping

Data-base report generators

Name: David Wyatt

Address: 415 Warren Road
Ithaca, NY 14850

Telephone: (607) 257-1894

Electronic Address: BITNET: T.RF4J@CRNL20A

Affiliation: Professor of Southeast Asian History
Cornell University

Areas of Expertise: Southeast Asian languages, especially Thai and Lao
Bibliography, especially Southeast Asian

Ideas/Interests: A good tool for dynamic keyboard remapping (instant changes within application)
A good report generator for Microsoft File

Name: Dennis Gail Abbey

Landscape architecture

Address: Room 302, Design Building
School of Landscape Architecture
College of Design
Baton Rouge, LA 70803

Telephone: (504) 388-1434

Affiliation: Louisiana State University
School of Landscape Architecture

Areas of Expertise: Landscape analysis and design

Ideas/Interests: Design templates for landscape architecture
Uniform specification system

Name: Michael T. Ganci

Courseware applications

Address: Informatics Group, Inc.
80 Shield Street
West Hartford, CT. 06110

Authoring systems

Staff development

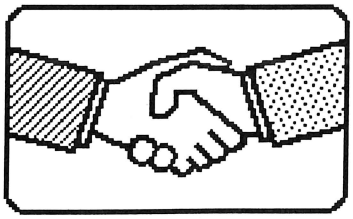
Telephone: (203) 724-4040

Beta testing

Affiliation: Consultant to several school systems, businesses, and municipalities

Areas of Expertise: Teaching and training
Authoring systems
Instructional design
Courseware development
Writing

Ideas/Interests: Courseware development collaboration
Authoring-systems conversion
Staff development seminar
courses consultation
Beta testing



Laser discs

Expert systems in education

Name: Peter A. Johnson

Address: 97 Winthrop St.
Augusta, ME 04330

Telephone: (207) 622-2535

Affiliation: Student at Bentley College
Brook 105-Bentley
Waltham, MA 02254

Areas of Expertise: Enthusiasm
Learning

Ideas/Interests: Laser-disk RAM and ROM
Expert systems in education

Communications

Legal expert systems

Computer-aided instruction

Social impacts

Name: Jennifer Bankier

Address: Dalhousie Law School
Halifax, Nova Scotia, Canada
B3H 4H9

Telephone: (902) 424-2114

Affiliation: Dalhousie Law School Associate Professor

Areas of Expertise: Computer conferencing
Law and computers
Social impact of computers
Computer applications in law schools

Ideas/Interests: Computer communications
Use of Macintosh in law firms
Legal expert systems
Computer-aided instruction
Social impact of computers

Social skills training

Human sexuality

Abnormal psychology

Name: Edward Zuckerman, Ph.D.

Address: 6307 Aylesboro Avenue
Pittsburgh, PA 15217

Telephone: (412) 367-1404
(412) 521-1057

Affiliation: University of Pittsburgh
Student Health Service
Wellness Resource Center
Brackenridge Hall

Areas of Expertise: Clinical psychology, therapy, marriage counseling, cognitive therapy
Rational emotive therapy, token economies, treatment of depression
Human sexuality, therapy, sexual politics, eroticism

Ideas/Interests: Social skills training, (dating conversation, assertion)
Teaching human sexuality to college students
Using a computer to teach abnormal psychology in college

Name: David Willcox

Address: University of Illinois at Chicago
Department of Chemical Engineering
810 S. Clinton
Chicago, Il 60607

Telephone: (312) 996-0848

Affiliation: Assistant Professor, Chemical Engineering

Areas of Expertise: Teaching
Programming
Data acquisition and analysis
Catalytic reaction research

Ideas/Interests: Development of a gas chromatography workstation
Chemical models for catalytic surfaces
Animation of chemical reactions
Undergraduate curriculum development in ChE

Gas chromatography

Chemical models

Chemical animations

Curriculum development

Name: Marvin L. De Jong

Address: Department of Mathematics-Physics
The School of the Ozarks
Pt. Lookout, MO 65726

Telephone: (417) 334-6411 ext 234

Statics

Laboratory devices

	Affiliation:	The School of the Ozarks
	Areas of Expertise:	Teaching math and physics Apple II interfacing 6502 assembly language Writing
	Idea/Interest areas:	Macintosh programs to solve statics (equilibrium) problems, especially trusses and shear-force bending-moment problems. Inexpensive laboratory interfacing devices.
Medical CAI ***Videodiscs*** ***Patient data/Rx systems***	**Name:**	Scott A. Iteen
	Address:	1793 7th Ave., Apt. #4 San Francisco, CA 94122
	Telephone:	(415) 665-7934
	Affiliation:	University of California, San Francisco School of Pharmacy
	Areas of Expertise:	Pharmacology Pharmacokinetics Clinical pharmacy Computer-based literature searches
	Ideas/Interest Areas:	Medical CAI Macintosh/Videodisc-interactive CAI Computer-based patient data/Rx system to allow increased interaction among health professionals
Programming ***Interactice media*** ***CD-ROM applications***	**Name:**	Robert Swierski
	Address:	125 East Encanto Drive Tempe, AZ 85281
	Telephone:	(602) 894-1390
	Affiliation:	College of Education Arizona State University Faculty associate and doctoral student

Areas of Expertise:

Teaching, training (including Macintosh applications)
Instructional design
Authoring systems on the Macintosh
Course development
Computer-based instruction

Ideas/Interest Areas:

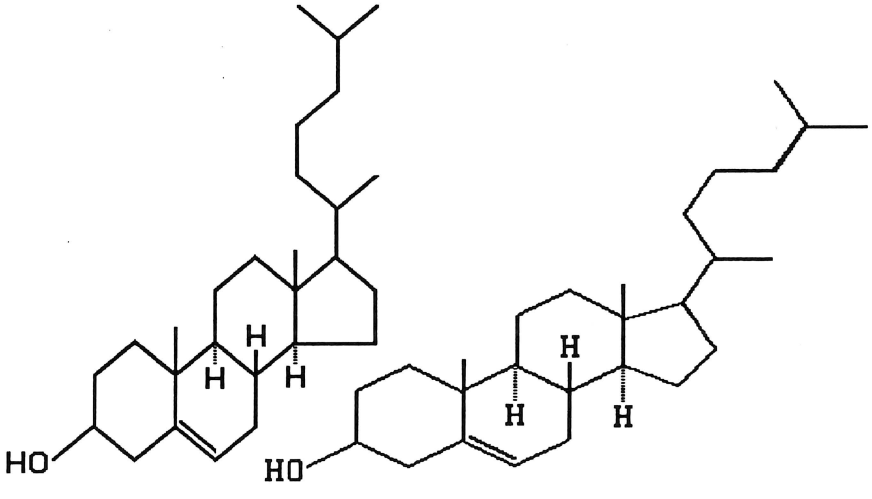
All of the above
Programming on the Macintosh
Interactive media
Automated instructional design research
Graduate education including on-line coursework
CD-ROM applications in higher education

Organic Fonts 1.1 Update

In the spring of 1986, Modern Graphics released **Organic Fonts**, a set of fonts designed to produce chemical structures. The three fonts included produce building blocks with single keystrokes, which may then be assembled into the desired structures. Many simple structures may be produced in word-processing programs, and virtually any but the most exotic structures may be produced in graphics programs.

Many simple structures can be produced within word-processing programs.

After the initial release, Modern Graphics received requests from users to include a new set of rings for pentagonal and hexagonal representations of 5- and 6-membered rings. An update, Version 1.1, was released, including the requested rings; also included is a guide to LaserWriter printing. Now, both of the following sterol structures may be constructed, whereas previously only the structure on the left could be constructed:



Modern Graphics has also received comments attesting to the fact that creation of structures directly in a word-processing program is not straightforward. Although many structures may be created, the font designed for use in word processing makes use of nonadvancing characters to allow placement of substituent groups on rings already typed; use of this font requires practice, and may not be worth the learning time for many. However, this font may also be used to produce building blocks in graphics programs, providing a method for producing smaller-size structures. Also, with the variety of desk accessories available for accessing and producing graphics, larger memory computers, and Switcher™ software, the capability to produce a structure directly within a word-processing program is not of great importance.

Modern Graphics mailed its 1.1 update free of charge to all registered owners who purchased the 1.0 version. The company will maintain this policy for future updates, as well.

Organic Fonts, $79.95. Academic discount, 10 percent; volume discounts available.

For further information, contact:

Modern Graphics
P.O. Box 21366
Indianapolis, IN 46221
(317) 856-8609

Drexel University: Publication Development Center

This center is devoted to publication design, imaging, printing, and publishing.

In the last two years, Drexel University's Nesbitt College of Design Arts has made a bold entrance into the area of printing and publishing education by creating several new academic programs, including a Publication Management Program and a Printing Technology Management Program. In conjunction with this effort, the college has developed a center devoted to publication design, imaging, printing, and publishing.

The Publication Development Center (PDC) was started by Compugraphic Corporation's donation of an MCS 100/8600 imagesetting system with four workstations, preview, film processor, flat-bed scanner and all appropriate software and maintenance contracts. The center serves as an academic laboratory, devoted to teaching students about the publishing industry through the development and publication of newsletters, funded grant research reports, refereed journals, and other campus-based research and information publications. Students from the colleges of Design Arts, Business and Administration, Information Studies, and

Humanities and Social Science are involved with the management, development, and prepress production of PDC efforts.

Several courses are available.

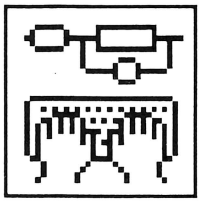
Currently, the college offers Computer Image Generation and Telecommunications, Graphic Design, Publication Layout, and Small Publication Management courses through its printing and publishing education programs. There are no programs like this in the area, and there are few similar programs in the nation. The programs and the PDC have received enthusiastic industry response, including cooperative education and graduate education placements, and major equipment donations from such print-industry giants as Compugraphic, DuPont, and Agfa.

The center will be a laboratory for the use and advancement of electronic publishing technology.

The college's intent is to continue to expand the scope of the Publication Development Center as a laboratory for the use and advancement of electronic publishing technology. A more ambitious goal is to develop a new model for publication management technology—to go beyond simple desktop production of individual documents to a microcomputer-based integrated-systems approach that meets total corporate office management and publishing needs. As envisioned, the system will take full advantage of productivity software, addressing all of the needs of electronic publishing: writing, editing, prepress production, office management, accounting, and telecommunications. The Macintosh (particularly its networking capabilities), Apple's obvious commitment to the electronic publishing market, and Drexel's commitment to the Macintosh will provide a solid basis for the effort.

One major goal of the PDS project will be to create a model microcomputer-based system for higher-education administration. The Macintosh systems in the key record-keeping offices of the college will be networked to allow shared file access, use of electronic mail, and data control for administrative management and generation of documents. This networking—offering shared access to data bases and records—will allow departments and offices to produce forms, reports, and other documents that can be customized to meet their needs and easily modified to meet changing needs, using the Publication Development Center.

Technical Information



The **Technical Information** section will present articles that are of a more technical nature than those in other parts of *Wheels for the Mind*. Here is where you might find information about new upgrades, technical references, and programming topics.

This particular issue contains a description of how to develop Macintosh applications, repairing MacWrite documents, and Smalltalk-80.

MacApp: A New Approach to Macintosh Application Development

Macintosh applications have a reputation for being easy to use, but until recently, difficult to program. Object-oriented programming, however, makes Macintosh application development easier. And Apple has developed an object-oriented system, called MacApp, to help programmers speed development time while they maintain interface consistency.

MacApp reduces development time.

MacApp reduces development time by giving the programmer a framework in which the general-purpose control structure is already written. MacApp is a real Macintosh program—just an empty one, waiting to be expanded into a specific application. (In fact, its full name is "MacApp: The Expandable Macintosh Application.")

The quality of an application can be improved.

In addition to reducing programming time, MacApp actually improves the quality of most applications. One reason for this is that MacApp performs extensive error checking and provides detailed alerts for user error messages. Another is that the MacApp code has been tested during its two years of development, while an application created from scratch must be debugged from scratch.

User interface details are automatically handled.

Because MacApp is an actual application, it automatically handles the common user-interface details, such as moving, resizing, and scrolling windows. It does not, however, handle application-specific actions, such as drawing within a window. But MacApp's object-oriented structure provides a convenient way for programmers to implement specific applications. Instead of editing the source code, they simply define new object types that override methods in MacApp and inherit the standard behavior. This means that code common to many applications does not need to be rewritten each time.

Object-oriented programming was used in MacApp for several reasons. The first relates to structure—the ability to create the framework just described. Another relates to programming style. Programs written in an object-oriented language are easier to maintain because the interactions between objects are through well-defined method interfaces; and the code can be more easily reused and extended through subclassing and inheritance.

MacApp is implemented using Object Pascal.

MacApp is implemented using Object Pascal, a version of Pascal that includes extensions for object-oriented programming. The object-oriented extensions were integrated with standard Pascal, so that programmers who already know Pascal can easily learn the language. Apple's new Macintosh Workshop Pascal product includes an Object Pascal compiler.

For more information about MacApp, consult the following sources:

BYTE magazine, August 1986 (This issue's theme is object-oriented languages.)

Kurt J. Schmucker, *Object-Oriented Programming for the Macintosh*, Hayden Book Company, 1986.

Macintosh 68000 Development System, Version 2.0

Users can develop sophisticated Macintosh applications.

The Macintosh 68000 Development System (MDS) is a complete assembly-language programming system for creating Macintosh software. It enables software authors to develop sophisticated Macintosh applications directly on any Macintosh system with at least 512K of RAM.

The Macintosh 68000 Development System can be used in conjunction with *Inside Macintosh* (a comprehensive technical description of the Macintosh ROM and user-interface considerations and programming guidelines) available from Addison-Wesley.

The basic features of MDS include:

- A window-based text editor.
- A 68000 assembler for translating Motorola 68000 assembly-language source files into machine-language modules.
- A linker for connecting groups of machine-language modules.
- An executive processor for automating the editing, assembler, and linker functions.

- A set of debuggers, including a window-based debugger.
- A set of system-definition files for symbolically referencing the Macintosh ROM routines and data structures.
- A resource compiler for defining application-specific resources.
- A number of useful utilities and detailed example programs.

Version 2.0 is now available.

Now, Version 2.0 of the Macintosh 68000 Development System (MDS) has been released. MDS 2.0 features many improvements over MDS 1.0, including:

- Full support for HFS, the Hierarchical File System used with the Macintosh Plus, the Macintosh 512K Enhanced, and many hard disks including Apple's Hard Disk 20.
- A new linker that strips out dead code, allowing users to create smaller applications.
- An improved editor that supports Undo and up to 9 windows.
- A path manager that allows users to customize the way MDS applications search for files in HFS folders.
- New equate and trap files that allow full access to the 128K ROM found in the Macintosh Plus and the Macintosh 512K Enhanced (as described in the recently published *Inside Macintosh Volume IV*).
- Debugger support for all 128K ROM calls.
- Corrections to a number of bugs found in MDS 1.0.

MDS 2.0 software is shipped on three single-sided Macintosh disks and will work on any Macintosh or Macintosh XL with at least 512K of RAM. Macintosh 128K owners who do not choose to get a memory upgrade can continue to use MDS 1.0.

If you don't already have MDS, your dealer can order MDS 2.0 (Apple part number M0524A). There will be a "Version 2.0" sticker on the new boxes. *MDS 2.0 is also available to universities in a 10-pack version containing 10 copies of the software and documentation.* (Each copy comprises three disks and two MDS manuals.) The 10-packs do not include the MC68000 manual (which is available from Motorola) or the optional debugging cable.

If you already own MDS 1.0, you can get an MDS 2.0 upgrade package, which will include the same three disks as the regular MDS 2.0 package, along with a 64-page update manual explaining the differences between the two versions (you will still need your old manual).

To obtain the upgrade package, back up your MDS Disk 1 and send the *original* disk (as proof of purchase) along with $25 (plus sales tax for California residents) and the following form to:

Apple Computer, Inc.
MDS 2.0 Upgrade
467 Saratoga Avenue, Suite 621
San Jose, CA 95129

(This address is only for orders from within the U.S. Outside the U.S., contact your authorized Apple representative.)

Upgrade orders must be postmarked by February 28, 1987.

MDS 2.0 Upgrade Package Order Form

MDS 2.0 Upgrades: ____ @ $25 each _____ (Include 1 MDS Disk 1 for *each* upgrade ordered)

Shipping and Handling No charge

California residents, add 7% sales tax _____

Total amount enclosed _____

Make check or money order payable to Apple Computer, Inc.

Include payment with orders. S & H included. Orders are shipped by U.P.S. Federal Express is optional but doesn't give processing priority.

My Federal Express billing number is:

_____

Ship order to (please print clearly):

Name _____

Company _____

Street Address _____

City _____ State _____ Zip Code _____

Phone (____)_____

Please allow 4 to 6 weeks for delivery.

Object Pascal

Information on Object Pascal now available.

Apple has collaborated with Professor Niklaus Wirth to design an object-oriented extension of Pascal called Object Pascal. *Object Pascal extensions are included in the new MPW Pascal compiler.*

The specifications of Object Pascal have been published in *Structured Language World*, Volume 9, Number 3, in an article by Apple's Larry Tesler titled "Object Pascal Report" (this is now also available as *Apple Technical Report #1*). We encourage others to implement the language on different machines. Documentation that explains how Object Pascal has been implemented on the Macintosh will soon be available by writing to:

Tools and Languages Group
Apple Computer M/S 27S
20525 Mariani Avenue
Cupertino, CA 95014

Repairing MacWrite Documents

Have damaged MacWrite documents? Read on.

The following descriptions assume the availability of the programs: MacWrite 4.5, Microsoft Word, ReadMacWrite, and FileTricks. There is a disk prepared for this purpose containing MacWrite 4.5, ReadMacWrite, and FileTricks, along with a desk accessory, SetFile, and this document. The program FileTricks can do the following:

- Extract ASCII text from any file.
- Duplicate a file.
- Extract Compressed text (MacWrite 3.0 or later) from a MacWrite file.
- Change fonts within a MacWrite document (non-LaserWriter to LaserWriter).

If Compressed text needs to be extracted from a file not recognized as a MacWrite file, then the "Creator" and "Type" attributes of the file need first to be changed to MACA and WORD, respectively. This can be done via the SetFile desk accessory or some other file utility such as FEdit.

RAM-based MacWrite (Version 2.2 or earlier)

MacWrite documents produced on MacWrite Version 2.2 sometimes become unreadable. There are two circumstances that commonly cause this:

1. A document was to be printed, the rasterized copy was being produced, and the *disk space* was exhausted. Sometimes MacWrite 2.2 does not detect disk space exhaustion, and the document is trashed. (Sometimes even the System file on the disk is trashed.)

2. A document is being written and the *memory space* becomes exhausted. Sometimes MacWrite 2.2 does not detect the problem, and again, the document is trashed.

In either case, the control codes on the document file get corrupted, and MacWrite 2.2 can no longer read the document.

Repair Methods

1. Try to read the document with a disk-based version of MacWrite(Version 4.5) or Microsoft Word:
   a. Open MacWrite (or Word).
   b. Close the Untitled document.
   c. Select Open from the File menu and choose the file that needs attention.
   d. Click OK, allowing it to be converted to the newer format.
   e. If this succeeds, you gain access to the file.
   f. If you want to return to RAM-based MacWrite (Version 2.2), then save the document as "text only." It can then be read by MacWrite Version 2.2., but be aware that all Font and Style information will be lost.

2. Extract the text (ASCII) from the file using FileTricks (formatting and pictures will be lost):
   a. Open the FileTricks program.
   b. Select the Extract ASCII Text option.
   c. Select the MacWrite 2.2 document file from the list (scroll if necessary).
   d. Name a destination file (on same disk or another if you like).
   e. Wait for process to be completed (you can cancel by pressing Control-period).
   f. Click on the Quit button.
   g. Open the MacWrite 2.2 application.
   h. Close your Untitled document.
   i. Select Open from the File menu.
   j. Choose the modified document just created.
   k. Remove the garbage and use the rest.

Disk-based MacWrite (Version numbers 3.0 and higher, usually 4.5)

The disk-based versions of MacWrite use a compression scheme to store documents. The process involves compacting text so that frequently used letters take up less space than those less often used. The major consequence of that is that the document cannot be read by other programs. Sometimes MacWrite gets confused and crashes before it gets a chance to write the document correctly to the disk. In subsequent attempts, MacWrite can no longer read it.

Repair Methods

1. Try to read the document with Microsoft Word:
   a. Open Microsoft Word.
   b. Close the Untitled document.
   c. Select Open from the File menu, and choose the file that needs attention.
   d. Click OK, allowing it to be converted to the Word format.
   e. If this succeeds, you gain access to the file.
   f. If you want to return to MacWrite, then save the document as "text only." It can then be read by MacWrite, but be aware that all Font and Style information will be lost.

2. Use a text-extraction program such as ReadMacWrite: (This program expects paragraphs to be intact. Formatting and pictures will be lost.)
   a. Open the ReadMacWrite program.
   b. Select To File and Screen from the Read menu.
   c. Choose the document to be read.
   d. Name a file to receive the results.
   e. You will see a lot of junk flash by. Hopefully, most of it will be readable text.
   f. Open MacWrite.
   g. Close the Untitled document.
   h. Select Open from the File menu.
   i. Choose the document just extracted.
   j. Remove the garbage and use the rest.

3. Extract the Compressed text from the file using FileTricks: (This program ignores paragraph pointers. Formatting and pictures will be lost.)
   a. Open the FileTricks program.
   b. Select the Extract Compressed Text option.
   c. Select the MacWrite document file from the list (scroll if necessary).
   d. Name a destination file (on same disk or another if you like).
   e. Wait for the process to be completed (you can cancel by pressing Control-period).
   f. Click on the Quit button.
   g. Open MacWrite.
   h. Close your Untitled document.
   i. Select Open from the File menu.
   j. Choose the modified document just created.
   k. Remove the garbage and use the rest. Note that paragraphs are probably out of order.

OH NO, MY FILE!!!!!!

By Joseph F. Buchanan, UUCC

"So, your file will not load. What did you say? You don't have a backup? Tsk, tsk, tsk!! How did you get yourself into such a fix?" (Of course, you need the file recovered immediately....)

What is a poor user to do? And what is a poor "MacGuru" to do, with an unhappy queue at his door? (Or is it a heap or stack? It depends upon the time of day.)

First, let's get you properly equipped. (I'm assuming that you know how to use all of these items.)

The Arsenal

- An extra disk or two (single- and double-sided)
- MacTools 5.4 (or so)
- FEdit Plus 1.04
- Copy II Mac 5.4 (or so)
- ReadMacWrite (Scott Gillespie, Reed College)
- Microsoft Word
- QUED 1.3
- FileTricks (Joseph F. Buchanan, University of Utah)
- A small paperclip, partially unbent (for unwilling disks in a 400K drive)
- A large paperclip, or a small piece of reinforced steel (for more advanced drives)
- A large handkerchief (for sobbing users)
- A Macintosh Plus, a hard disk, common sense, compassion, and so forth (optional, but recommended)

Procedures (Please note that the following is not for the faint of heart.)

1. Calm the user (hysterics are common).
2. Rephrase the user's assessment of the situation (translate to English if necessary).

3. Determine the type of file to be recovered. Classify as follows:

   M—MacWrite document (Version 3.0 or higher)
   A—Text-type document (stored in ASCII)
   O—Any other

   Be aware that most files can be treated as Type A. For example, Microsoft Word keeps its text in ASCII. Most spreadsheets and data bases keep the actual data in ASCII. The special formatting and organizational information is not in ASCII and would probably be lost anyway. Also, at present, you can probably write off anything in PICT form unless the next few steps work.

4. Try the disk in a different machine. Sometimes there might be a calibration problem in the drive, causing misreads. If this works, you become an instant hero (or heroine), but be sure to suspect a problem with the original drive and have it tested.

5. Try rebuilding the desktop file. Two methods are possible:
   a. Insert the disk (after reboot) and hold down the Option and Control keys through the entire process.
   b. Use a utility like MacTools to delete the desktop file, which will cause the Finder to rebuild it.

6. *Make a copy of the disk before trying anything else.* Use the copy in all subsequent attempts at recovery. Usually you will need Copy II Mac to copy the disk, using bit copy. Sometimes this in itself will produce a recovered disk, but that is not likely. Because you are working with a copy, you can always get back to the original if something goes awry. Also, users can try someone else if they don't like your work (also not likely).

7. Is the disk itself unreadable? If so, repeat Step 1, and then skip the next sections down to the Disk Recovery section.

8. Has the file been deleted, or is it not accessible by any means? If so, go to the Disk Recovery section.

9. Type M:
   a. Try reading it with Microsoft Word. It might work.
   b. Try reading it with ReadMacWrite. It might work, but you will lose font and style information, formatting, and pictures.
   c. Read it with FileTricks, Compressed format recovery. It will read the file, but you will lose font and style information, formatting, and pictures. Also, paragraphs will be out of their proper order.

10. Type A:
    a. Try reading it with QUED. It might find it. You might have to zap gremlins.
    b. Read it with FileTricks, ASCII format recovery. It will read the file, but you will probably see a lot of garbage.

11. Type O:
    a. Good luck! Try FileTricks reading it as ASCII. It might look like pure garbage occasionally sprinkled with good stuff. The user will need to decide if he or she would rather clean up the garbage or start from scratch (sometimes the cleanup work is more time-consuming).

12. If nothing good comes from the preceding steps, you could try the methods listed in the next section, but don't hope for much.

Disk Recovery

1. Determine the disk formatting:

   - 400K MFS—standard single-sided floppy
   - 400K HFS—unusual for a single-sided, but possible
   - 800K MFS—double-sided "flat"
   - 800K HFS—standard double-sided hierarchical

   Note: Do *not* believe the writing on the disk. Many single-sided disks are being used as 800K. This might even be the cause of the problem.

2. Use MacTools to look at the disk:
   a. If it will load the disk and show your file(s), then try copying it (them) to another disk.
   b. If it will load the disk and your file(s) do not show, then try to UnDelete files.
   c. If it won't load the disk, select Repair and then hope for the best. If you get some files, go to Step 7.

3. If Step 2 failed, you will probably need to restore your disk (see Step 6 in the previous section). Then continue to Step 4 or 6 in this section.

4. 400K HFS or 800K HFS:
   a. Prepare a blank 400K MFS or 800K MFS disk.
   b. Proceed to Step 5.

5. Run Copy II Mac and do a sector copy *(without* formatting) of sectors 0 through 2 from the blank MFS disk to the troublesome HFS disk. This will trick the system into thinking it is working with an MFS volume. It will also change the name of the disk. Proceed to Step 6.

6. 400K MFS or 800K MFS:
   a. Run FEdit Plus, and do an Open Volume.
   b. Insert the disk, and select Recover Deleted Files from the Special menu.

7. If this worked, then files will be found, but their names will not be familiar. You will also need to look at the individual files with FEdit Plus and restore the Creator and Type fields of each file. For example, a MacWrite file should be of Type WORD, Creator MACA. You can usually recognize a MacWrite file by its small Resource fork containing the days of the week and month names and the data fork beginning with Hex 0006.

8. If the preceding steps failed, try reading the disk with FileTricks. To recover an entire disk, you must run FileTricks from a hard disk or from a disk in the external drive, and the disk must have plenty of room to receive a lot of junk. FileTricks will read the disk starting with Sector 15 (past the directory stuff) extracting either ASCII or Compressed (MacWrite 4.5 text). Choose Compressed only if you expect MacWrite format (otherwise choose ASCII). Most of what you see will be garbage. An entire 800K disk recovery using Compressed can take more than five hours and yield over a megabyte of junk. You can skip ahead in jumps of 1 to 9 blocks (512 bytes) and can terminate using Control-period. FileTricks will show a count of consecutive spaces, because zeros in Compressed form convert to spaces and may really be vast amounts of initialized disk. Again, you must ask yourself if this is really worth it, because the junk might be too vast to work with. This last step is definitely a last-resort process, but can be helpful. Also, FileTricks will ignore any I/O problems, so you might get left in a "hung" condition that requires a reboot of the machine.

9. If you reach this point, you might try the following with your disk:
   a. Reinitialize the disk. You couldn't get anything from it anyway, so at least try to make it useful. If it won't initialize, try to "degauss" it (large magnetic field), then try to initialize it again. If that still doesn't work, you can:
   b. Cry.
   c. Give it to an NRA friend for target practice.
   d. Use it as a drink coaster.
   e. Mount it on a plaque and put it on your wall as "the one that got away."
   f. Use it to stabilize a wobbly table.
   g. Paste a picture of your family on it and set it in the front of your disk organizer.
   h. Slice off a corner and try to sell it as valuable software that has been "reduced."
   i. Put a clock mechanism in the center of it, frame it, and sell it as a unique clock.
   j. Use it as a window scraper for your car in the winter.
   k. And so on.

Smalltalk-80 for the Macintosh, Release 0.3

What Is Smalltalk?

The programming environment includes both the language and some software utilities.

Smalltalk-80 for the Macintosh is a uniquely powerful, integrated object-oriented programming environment. Part of this environment is an object-oriented language called Smalltalk-80. Another part is a large library of powerful software utilities. The Smalltalk environment includes incremental compilation, source-level debugging, intelligent program editing, automatic storage management, run-time efficiency analysis, program change management, and more. When we refer to Smalltalk, we are talking about all three of these things: the language, the utilities, and the programming environment.

New Features of Version 0.3

- Fast update of windows, because Version 0.3 saves bit maps of underlying windows.
- Many new editor features designed to help the programmer.
- Support for the Hierarchical File System (HFS) of the Macintosh Plus.
- Improved support for Toolbox calls, including inspectors for Pascal format records in the Toolbox heap.
- Friendlier browser interface.
- A Versions command that allows you to see and revert to previous versions of a method.
- Underlying windows are easier to find.
- Many bugs from previous releases have been fixed.

Other Information About Smalltalk-80 for the Macintosh

At least 1 megabyte of memory is needed.

Smalltalk will run on all Macintosh computers with at least 1 megabyte of memory and a hard disk or one double-sided (800K) disk drive. This includes the Macintosh Plus and the Macintosh XL.

Smalltalk performance is generally measured relative to Smalltalk running on the Xerox Dorado. Smalltalk 0.3 performs at roughly 14.5 percent of a Dorado on a standard Macintosh Plus and at 57 percent of a Dorado on a Prodigy-4 Macintosh by Levco.

Automatic spelling correction is available for variable and message names.

Programmers have access to any Macintosh Toolbox call that is made through a trap.

Smalltalk supports a 32K-entry object table. On a 1-megabyte system, users will have about 220 kilobytes and 13,500 object-table entries for their own use.

Differences Between Smalltalk-80 for the Macintosh and Xerox Versions

Smalltalk-80 for the Macintosh is based on Xerox Smalltalk-80 Version I licensed from Xerox in an early collaboration on Smalltalk development. Apple developed its own interpreter and has made many improvements to the image since Version I. Xerox has independently improved Version I, which it now offers for general licensing as Version II. Smalltalk-80 for the Macintosh and Xerox Version II differ, but the Smalltalk language is still compatible and most of the kernel programming tools are similar.

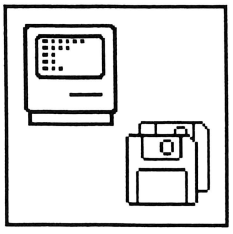
Availability

Smalltalk-80 is available through APDA.

Smalltalk-80 for the Macintosh is available through APDA (the Apple Programmer's and Developer's Association) as an unsupported product. Version 0.3 became available December 1, 1986. It costs $50, comes on seven single-sided disks, and includes a manual. Users who don't know Smalltalk will also need to purchase a book.

An institutional license is available that allows an institution of higher education to make copies of any of the Smalltalk releases available through APDA for use within the institution. The cost is $50.

Macintosh Products



The **Macintosh Products** section will feature brief descriptions of products that are of interest to the academic community. Some of the products will be specifically concerned with the theme of the current *Wheels for the Mind* issue, while others will be of more general interest.

Please send in your suggestions for features that might be of interest to those in the academic community.

MacNet Update

The MacNet project is dedicated to the integration of Macintosh personal computers and future Apple products into the Carnegie-Mellon University (CMU) campus network.

The project's goal is to enable connection of a single Macintosh personal computer, a cluster of PCs, or a network of PC clusters to the university computing network. Thus, the Macintosh systems will be able to share in the network resources available on campus, from file service to bulletin board access. Another goal is to promote the uniform interfacing of Macintosh computers to other networks, both in universities and commercially.

The goal is to interface Macintosh computers with all other computers on campus.

How It Works

Macintosh personal computers and peripherals will be arranged in an AppleTalk cluster. Several clusters of Macintosh computers may form a star-shaped network through an AppleTalk hub. The resulting network will attach to the campus internet via a gateway, which serves as a mediator between the two different networks.

The gateway hardware to be used in this scheme has been developed by Lawrence Butcher of CMU's computer science department. The software has been adapted from the "Seagate" code written by Bill Croft of Stanford University. The gateway provides routing between IP and AppleTalk DDP nodes, as well as AppleTalk/IP encapsulation. Initial important IP work was done by Tim Maroney and Mark Sherman.

MacNet will facilitate transparent file service for the Macintosh using a UNIX machine as the server. Other attractive features are print spooling to the LaserWriter or ImageWriter printer, mail service, and bulletin board service.

Three programs have already been developed.

Three programs, CUSTOMIZE, Telnet, and TFTP, have been developed to allow Apple Macintosh computers on an AppleTalk network to communicate with other computers on the campus TCP/IP internet. CUSTOMIZE manages a site-specific file that contains information for use by the other programs. The Telnet application allows the user to log onto any host system for which the user has an account, if that host provides Telnet service (as most TCP/IP hosts do). The TFTP (trivial file transfer protocol) program is used to transfer files between Macintosh computers, or a Macintosh and another internet site.

Equipment

Most MacNet applications require only a Macintosh 128K. The external file system software may necessitate a Macintosh 512K. In addition, a gateway is needed to interface the AppleTalk network to another IP network.

The IP applications mentioned are available now, and the sources have been posted to the usenet. Similiar gateway hardware and software are available commercially (from Kinetics, Inc). For more information on CMU-specific distribution, contact Rob Chandhok or Mark Brown at the contact address given.

Any MacNet software costs will be only to cover distribution expenses. The "Butcher Board" may be available in the near future, but the price is currently unknown.

The following section offers details on the MacNet connection—the hardware involved and the functions available.

Basic Communication

- Hardware
  —Butcher Board: AppleTalk to Ethernet Gateway (Kinetics)
  —MacDELNI: AppleTalk Hub (Blender Bus, Webber Hub)

- TCP/IP Software
  —Gateway: Seagate-style AppleTalk-to-IP communication
  —Telnet: remote terminal emulation
  —TFTP: file transfer

Services

- Transparent Services
  —Print spooling: LaserWriter and ImageWriter, using 4.2 UNIX machine as a spooler
  —File serving: Macintosh to 4.2 UNIX interface

- Campus Integration via Central Message Server
  —Mail: central post office, always open
  —Bulletin boards: central repository for large data bases

For further information, contact:

Center for Art & Technology
Carnegie-Mellon University
Pittsburgh, PA 15213
(412) 268-3454

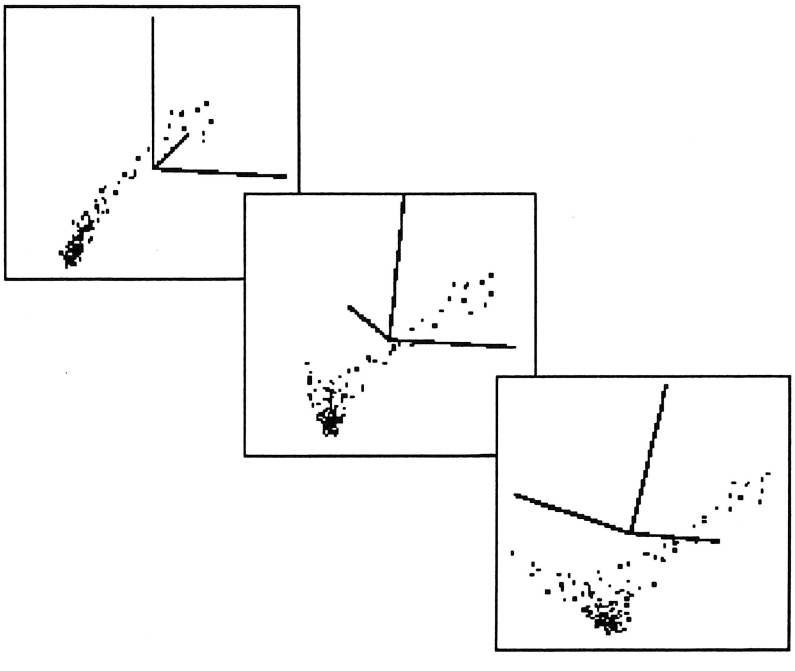
MacSpin—the Data Analyst's Wheels for the Mind

By Andrew W. Donoho, University of Texas, Austin
David L. Donoho, University of California, Berkeley
Miriam G. Donoho, University of Chicago

MacSpin establishes a totally new category of graphics software. It allows you to view and interact with data having three or more dimensions. Such data can occur in any field of study where three or more numerical variables play an important role. MacSpin has been used to look at data in many fields — ranging from astronomy and seismology to botany and urban planning.

MacSpin allows you to view and interact with your data.

MacSpin takes off from a familiar point of departure: the x-y scatterplot, in which (x,y) data (for example, height and weight, or age and income) are plotted against each other. With MacSpin, you can view x-y-z plots, rotating in real time to get a true three-dimensional perception of the structure of your data. Views of such rotating plots, a few moments apart, might look like this:



Rotating x-y-z plots

The rotation shows that a data set that, from some views might appear to be a simple ellipse, actually forms an L, when viewed appropriately. The data have to do with diagnosis of diabetes; the L tells us that there are two different kinds of diabetics. The three-dimensional scatterplot lets us see this immediately.

You can make movies of your data.

But MacSpin offers much, much more than three-dimensional scatterplots. With MacSpin you can make movies showing how your data vary as a function of a fourth variable (for example, time). You can interact with your data, identifying outliers, and correlating text and qualitative information with patterns you observe in the x-y-z display. In a very few minutes, you can see your data from a wide variety of viewpoints. You can identify trends, patterns, clusters, and outliers easily. And MacSpin works with the standard, easy-to-use point-and-click operations of the Macintosh mouse.

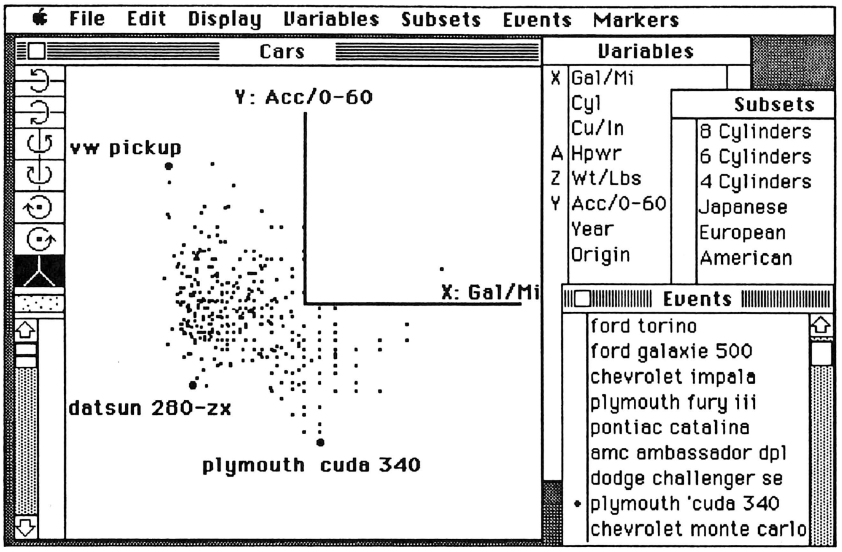
Users of MacSpin have said that analyzing data with MacSpin is "interesting, lively, and exciting" and that the program makes the process of understanding abstract data "intutive and immediate." The February 1986 issue of *MACazine* called MacSpin an "information observatory."

The MacSpin program was written by Andrew Donoho, a physicist; David and Miriam Donoho, statisticians, collaborated in the design and produced a 189-page manual describing the use of the program. This manual provides detailed descriptions of the application of MacSpin to seven interesting data sets from a variety of fields. These data are distributed with the program so readers can work through the examples as they go along.

An Application Story

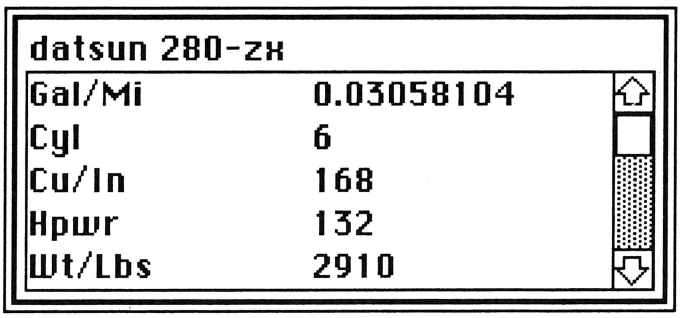
Looking at some actual data with MacSpin.

To illustrate the main features of the program, we can simply tell a story of its application. The data we'll look at consist of all the cars road-tested by *Consumer Reports* magazine between 1971 and 1983. The data will help us see how the auto industry has changed over the last decade. The names of the 418 cars are listed in the Events window on the MacSpin display (lower right): the portion we see includes the Plymouth Barracuda and Plymouth Fury III cars from the early 1970s. The Variables window (partially obscured by the Events window) shows the variables we have measured for each car; things involving performance (gallons per mile, seconds to reach 60 mph from a full stop), size (horsepower, weight, and so forth), and miscellaneous (model year, continent of origin).



The x-y plot window

X-Y plots. The view in the plot window shows all the cars (American, European, and Japanese) in an x-y plot, with x=Gal/Mi (fuel usage per mile) versus y=slowness (as reflected by seconds from 0 to 60). The points represent individual cars. By moving the cursor to a point and clicking, we can find out its identity. The point at the upper left (slow but economical) is a VW pickup; the point in the lower right (fast gas-guzzler) is a Plymouth Barracuda. If we then hold down the Control and Option keys, the full data record pops up. This shows us that the fast, efficient car in the lower left is a 1981 Datsun 280 ZX.

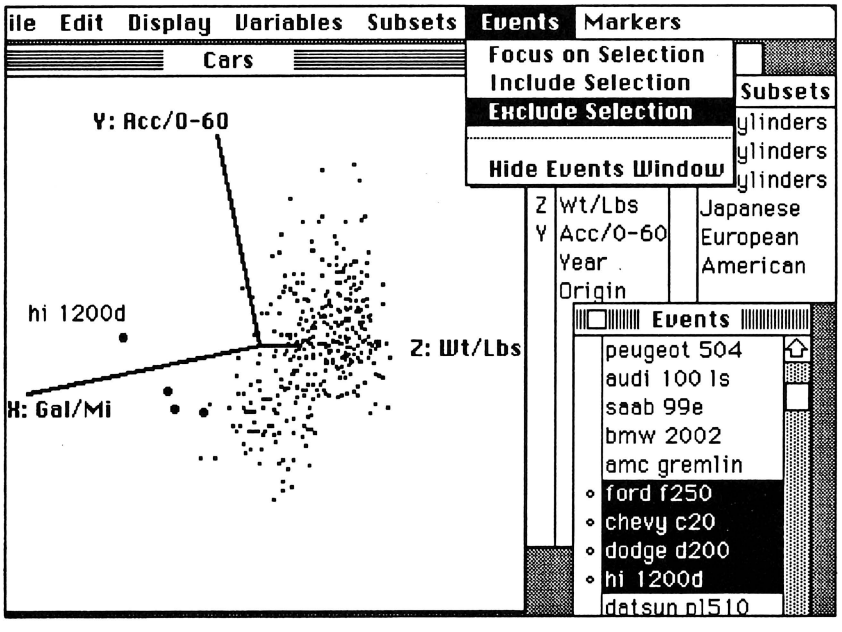


A data record of information

X-Y-Z plots. The x-y plot shows the general trend of the auto industry—what combinations of speed and economy are available. By rotating the plot, we can get an extra dimension into the display. We point at a rotation icon on the far right, and hold down the mouse button. The two-dimensional plot becomes a rotating three-dimensional plot, the previously hidden z-variable coming into play. When we do this for the Cars data, and bring z=weight into the display, we see a cloud of points rotating smoothly in space. The

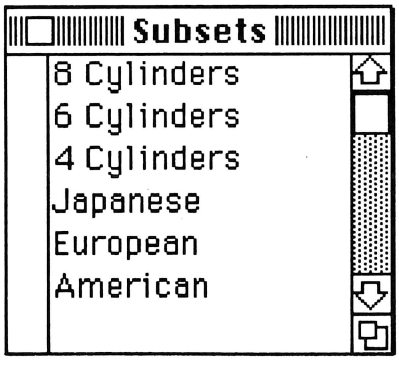
cloud is shaped like a sausage and shows the combinations of economy, speed, and weight produced in 1971-1983. As we rotate this, we notice a few interesting things. First, one point turns out to be an outlier. We stop the rotation and identify it; it is an International Harvester truck. When we scroll to the truck's name in the Events window, we see that at the same time, *Consumer Reports* also road-tested a few other trucks. By pointing at their names on the list, we can highlight them in the plot window. They turn out to be outlying also. Somehow trucks have slipped into a data base on cars! By choosing Exclude from the Events window, we can (temporarily) remove them from the display. The rotation has helped us identify and remove outliers.

x-y-z plots

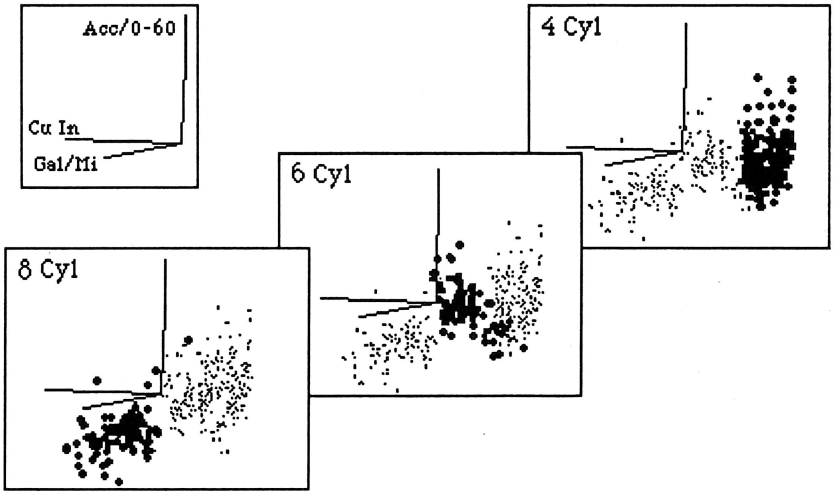


Highlighting subsets. Further rotation shows that the data consist of three clusters. Seeking for an explanation, we bring the Subsets window to the front. This shows some subsets of the data predefined (by us) as being interesting to look at.

Data subsets can be identified.

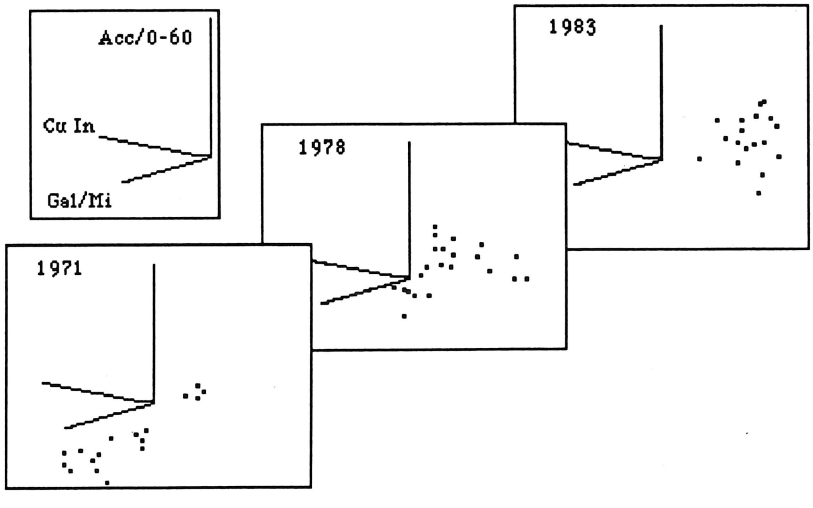


By pointing at the name of any subset, we can highlight its members on the display. When we do this, we see that the three clusters consist of 8-, 6-, and 4-cylinder cars, respectively. We could also highlight American, European, and Japanese subsets, in turn, and find out where they are on the display.



Clusters can easily be seen.

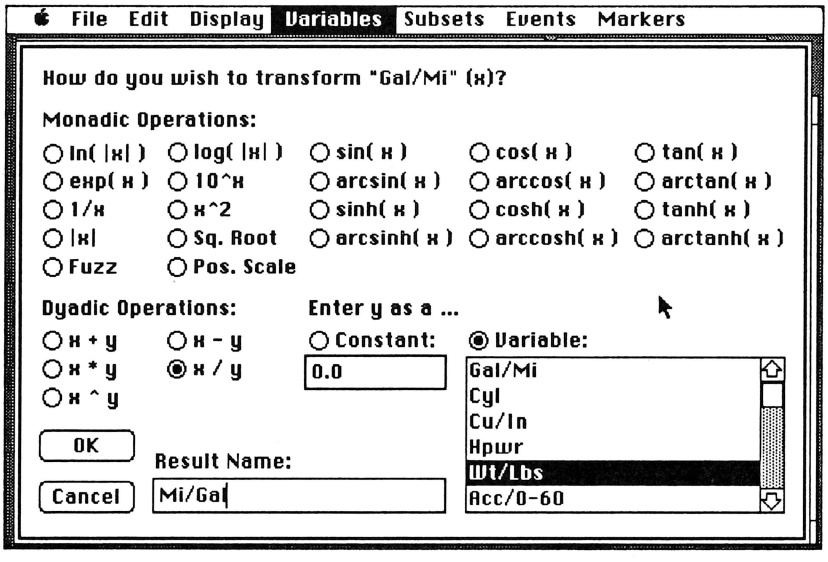
Movie time. We are interested in how the American auto industry changed over this period, so we highlight the American cars, and then select Focus from the Events menu. MacSpin then temporarily hides the imported cars from the display. It is time now for a movie of how the American auto industry changed from 1971 to 1983. We drag the Year variable to the scroll bar in the lower left of the MacSpin screen. This will let us scroll through the data, model year by model year. We begin at 1971. The cars made then are concentrated in the lower left of the display: fast, heavy, gas-guzzling cars. As we scroll smoothly forward, we see that the data drift systematically toward the upper left—toward slower, lighter, more economical machines.



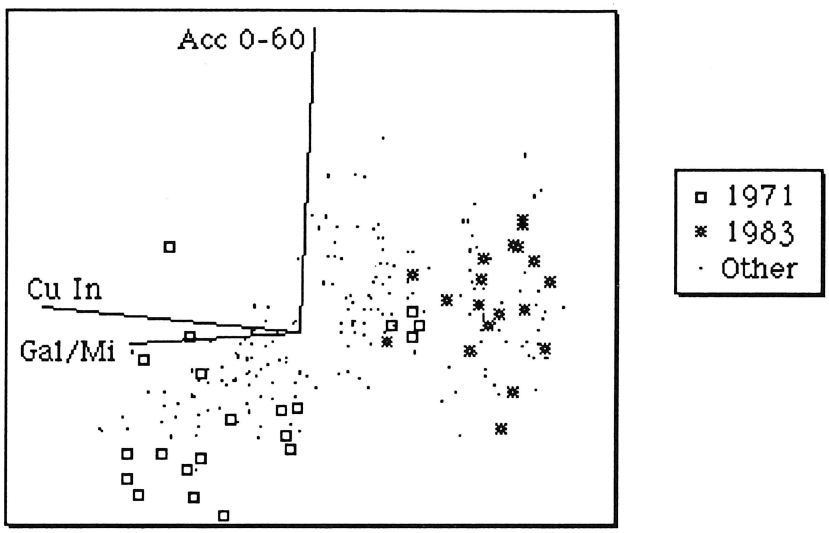
Animation helps identify groupings.

New variables. This is only the beginning of a full analysis of these data. At this point, we might create a new variable—such as a standardized measure of fuel efficiency (say gal/mi divided by weight) in which the effects of weight are taken out. Looking at plots with this new variable would give us insight into whether American cars got more efficient or whether they just got smaller over this period. Creating this new variable is as easy as pointing and clicking with the mouse. Features like this make MacSpin useful not just for displaying data but also for manipulating it to get the right display.

New variables can be created easily.



Printed image. You can even make a presentation graph at the end of your exploration. MacSpin lets you mark subsets with special symbols. You can use this to convey some of what the dynamic exploration showed you. The figure on the next page is derived from a MacSpin screen dump. MacSpin marked the 1971 model cars with a box, and the 1983 model cars with an asterisk. A screen dump was generated using the Command-Shift-3 sequence. The resulting image was cropped, and shadows and captions were drawn in, using MacPaint.



Wheels for the Data Analyst

The preceding story illustrates many of the important features of MacSpin:

- Rotation to show a third dimension.
- The ability to identify interesting points.
- The ability to highlight important subsets.
- Animation to look at a fourth variable.
- The ability to transform the data.
- The ability to mark subsets.

In short, MacSpin is a full-function tool for displaying multivariate data. The program can be used profitably with other Macintosh software, analyzing data coming from spreadsheets, data bases, and statistics packages. It can also be used (via MacTerminal™ and Switcher) as a way to inspect data bases on mainframes and minicomputers.

MacSpin is a full-function tool for displaying multivariate data.

This last point deserves emphasis: MacSpin can be used to open a "window" into a data base stored on some other computer. This may turn out to be the most frequent way MacSpin is used in a university environment. The Macintosh is used as a terminal to the other computer; the user logs in on the other machine and carries out a normal terminal session (for example, perhaps employing a data base program or a statistics package on that machine). The difference from an ordinary session comes when the user downloads some of the data into the Macintosh and inspects it with MacSpin. Rather than just seeing the numerical summaries and simple plots that the software on the other machine provides, the user supplements the analysis by interacting with the data more directly via MacSpin. The three-dimensional representation, animation, and highlighting combine to give the user, in a very few minutes, a number of insights to help guide his or her use of the data base or statistics program.

The interactive nature makes data analysis exciting.

MacSpin may have a big role in teaching students how to understand and use data. People who thought statistics dull and boring may find the dynamic, interactive nature of MacSpin encourages them to look at and think about data. MacSpin can also be used to help visualize statistical concepts—such as the ellipsoidal shape of multivariate "normal" data, and to let students compare real data visually with data obtained by simulation from the normal distribution. The applications to graduate courses on multivariate analysis are, of course, many.

History

MacSpin predecessors have been around for awhile.

Predecessors of MacSpin have been under development in research labs for about 15 years. The first system of this kind, PRIM-9, developed at the Stanford Linear Accelerator Center, ran on an IBM mainframe with a $400,000 display unit, and cost several hundred dollars an hour to use. Other, slightly less expensive to use, systems were built at Harvard and Stanford. Such systems served to show that dynamic graphics could be used to analyze data in many different fields—that this approach held a lot of potential. For example, PRIM-9 was used to analyze data from particle physics experiments, and it was also used to study the different diabetes syndromes. The PRIM-H system, built at Harvard, was used to inspect data on circadian rhythms of monkeys, on demographics of developing nations, and on clustering of remote galaxies.

Unfortunately, such systems were hidden away in labs and only a few people ever had access to them. Also, because they were used by so few people, issues such as ease of use and capacity for working with real data on an everyday basis were not addressed. The designers of these systems made movies and videotapes illustrating the tremendous potential of dynamic graphics, but they did not use their systems for everyday work.

MacSpin brings dynamic graphics out of the computer graphics laboratory, and onto the desktop of anyone with a Macintosh. And the program goes far beyond the research-based systems in ease of use and in applicability for everyday use.

MacSpin is being used today by scientists, engineers, doctors, and even some racetrack bettors. Anyone who needs to analyze multivariate data can benefit from viewing data *and interacting with it* using MacSpin.

More Information

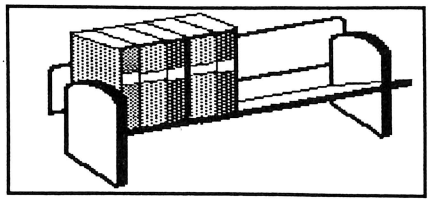
MacSpin is a commercial product available now. It consists of the MacSpin program, seven example data sets from a wide range of fields (astronomy to urban planning), and a 189-page manual that both tells how to use the program and explains data analysis. It is

available to educational institutions at a price of $79.95 from Wadsworth Advanced Books and Software, 555 Abrego Street, Monterey, CA 93940; (408) 373-0728.

For more information, contact Wadsworth or the program's developers:

D^2 Software
3001 North Lamar Blvd., Suite 110
Austin, TX 78705
(512) 482-8933

Resources



Macintosh Books

Macintosh Graphics and Sound
David Kater
Osborne/McGraw-Hill
276 pages

Macintosh Midnight Madness
(Utilities & Games in BASIC)
The Waite Group
Microsoft Press
420 pages, $18.95

Personal Financial Planning with The Home Accountant
Steve Adams
Arrays, Inc.
818-901-8828, 800-468-9222
248 pages, 17.95

The Apple Macintosh Book
(2nd Edition)
Cary Lu
422 pages, $19.95

Microsoft BASIC Programming for the Mac
Sharon Zardetto Aker
Scott, Foresman and Company
344 pages, $17.95

Short, Simple, and BASIC: Business Programs for the Macintosh
M. Robinson and T. Kepner
Scott, Foresman and Company
320 pages, $18.95

Microsoft File
Nancy Andrews
Microsoft Press
451 pages
$18.95

Mac at Work
S. Venit and D. Burns
Wiley Press, NY
228 pages, $17.95

MacGuide: The Complete Handbook for Macintosh
Leslie Smith
New American Library
277 pages, $14.95

Mac Power
Allen Munro
Scott, Foresman and Company
320 pages, $15.95

BASIC

Data Base

General Macintosh

Macintosh Books

Graphics, Art, Word Processing

The MAC Art Department
Tom Christopher
Simon & Schuster, NY
96 pages, disk, $39.95

The Printed WORD
David Kater & Richard Kater
Microsoft Press
Bellevue, WA
293 pages, $17.95
(Review: MACAZINE, Vol.3, No.2, p. 36)

MacGraphics for Business
Jerry Mar
Scott, Foresman and Company
194 pages, $17.95

Graphics for the Macintosh: An Idea Book
J. Grillo and D. Robertson
CBS Computer Books
Holt, Rinehart, and Winston
217 pages, $17.95

MacCats: 99 Ways to Paint a Cat with MacPaint
Floyd Flanagan
Scott, Foresman and Company
146 pages, $9.95

Miscellaneous

PostScript Language Reference Manual
Adobe Systems Incorporated
Addison-Wesley Publishing Company
319 pages, $22.95

AI Programming on the Macintosh
Dan Shafer
SAMS (A Division of Macmillan, Inc.)
290 pages, $24.95

PostScript Language Tutorial and Cookbook
Adobe Systems Incorporated
Addison-Wesley Publishing Company
243 pages, $16.95

Pascal

The Magic of Macintosh Graphics & Sound in Pascal
Bill Twitty
Scott, Foresman and Company
416 pages, $19.95

Learning Macintosh Pascal: A Guide for Programmers
J. Wikert & S. Davis
Scott, Foresman and Company
356 pages, $19.95

Scientific Programming with Macintosh Pascal
Crandall and Colgrove
Wiley Press
279 pages, $18.95

The Magic of Macintosh: Programming Graphics and Sound
William Twitty
Scott, Foresman and Company
314 pages, $19.95

Macintosh Books

Programming

Assembly Language Primer for Macintosh
Keith Mathews
The Waite Group
412 pages, $24.95

Mastering the Macintosh Toolbox
D. Peabody
McGraw-Hill Book Company
200 pages

Using the Macintosh Toolbox with C
Takasuka, Huxham, and Burnard
Sybex Books
559 pages

Programming in C on the Macintosh
Terry A. Ward
Scott, Foresman and Company
411 pages, $21.95

Programming the Macintosh: An Advanced Guide
William Twitty
Scott, Foresman and Company
374 pages, $19.95

Spreadsheets

Multiplan Mastery on the Macintosh
Andrew J. Townsend
Scott, Foresman and Company
260 pages, $18.95

Doug Clapp's JAZZ Book
Doug Clapp
Scott, Foresman and Company
272 pages, $17.95

Hands-on EXCEL
Danny Goodman
Scott, Foresman and Company
328 pages, $21.95

Making Your Macintosh EXCEL
Bill O'Brien
Scott, Foresman and Company
356 pages, $19.95

Articles of Interest—Business

Accounting

⌘Accounting Buyer's Guide	MACWORLD	Sept. 86	p. 88
Accounting: Plain and Simple	INFOWORLD	June 30, 86	p. 37
Accounting Software for Mac	MACUSER	Dec. 85	p. 52
Accountant's Choice	MACUSER	Dec. 85	p. 56
Back to Basics Accounting	MACWORLD	Dec. 85	p. 75
BPI General Accounting	MACUSER	Aug. 86	p. 33
Check Mark General Ledger	MACUSER	Dec. 85	p. 56
Dollars & $ense	MACAZINE	Vol. 3 No.1	p. 18
⌘Dollars & $ense	MACWORLD	Dec. 86	p. 183
Designing Forms	MACWORLD	June 86	p. 89
The Home Accountant	MACAZINE	May 86	p. 85
⌘Insight	INFOWORLD	Oct. 10, 86	p. 45

⌘Insi

Articles of Interest—Business

⌘Insight	INFOWORLD	Oct. 10, 86	p. 45
⌘Insight	MACWORLD	Oct. 86	p. 122
Maccountant	MACUSER	Dec. 85	p. 58
Mac OneWrite	MACUSER	Dec. 85	p. 58
Mac OneWrite	MACWORLD	Mar. 86	p. 123
MacProject	MACAZINE	June 86	p. 37
Micro Planner	MACAZINE	June 86	p. 37
⌘Micro Planner Plus	MACWORLD	Oct. 86	p. 146
Palantir—Accounting	MACUSER	Dec. 85	p. 114
Palantir Financial Series	MACAZINE	Vol.3 No.1	p. 13
Profit Stalker II	MACUSER	Aug. 86	p. 34
Project Management	MACAZINE	June 86	p. 37
Rags to Riches	MACUSER	July 86	p. 34
Rags to Riches—Accounting	MACUSER	Dec. 85	p. 114
Rags to Riches	MACAZINE	Vol.3 No.1	p. 14
Strictly Business—Accounting	MACUSER	Dec. 85	p. 114
⌘Turbo Maccountant	MACWORLD	Oct. 86	p. 156

Taxes

MacInTax	MACUSER	Jan. 86	p. 46
MacInTax	MACWORLD	Mar. 86	p. 88
MacInTax	MACAZINE	Vol.3 No.1	p. 64
Mac Tax Packages	MACAZINE	Vol.3 No.3	p. 25
Tax Wizard	MACUSER	Jan. 86	p. 48
Tax Wizard	MACAZINE	Vol.3 No.1	p. 64
Tax Wizard	MACWORLD	Mar. 86	p. 89
Tax Prep	MACUSER	Jan. 86	p. 48
Tax Prep	MACWORLD	Mar. 86	p. 90
Tax Plan	MACWORLD	Mar. 86	p. 90
Tax Master	MACWORLD	Mar. 86	p. 91

Miscellaneous Business

⌘Appointment Diary	MACWORLD	Dec. 86	p. 145
Consultant	A+ MAGAZINE	Apr. 86	p. 126
Corporate Computing	A+ MAGAZINE	Apr. 85	p. 78
Decision Map	A+ MAGAZINE	Apr. 86	p. 124
Decision Map	MACWORLD	Apr. 86	p. 102
Financial Cookbook	MACWORLD	Apr. 86	p. 132

Legal Billing	MACWORLD	Dec. 86	p. 200
Macintosh Data Exchange	MACWORLD	Dec. 86	p. 104
Macintosh in Corporate World	INFOWORLD	Oct. 13, 86	p. 15
MacProject	MACWORLD	Nov. 86	p. 189
MacProject	MACUSER	Mar. 86	p. 54
Mail Center	MACWORLD	Dec. 85	p. 82
Mail Manager	MACWORLD	Apr. 86	p.125
Market Pro	MACUSER	Nov. 86	p. 47
MegaForm	MACUSER	Feb. 86	p. 69
MindSight	MACWORLD	Apr. 86	p. 96
MindSight	A+ MAGAZINE	Apr. 86	p. 120
Micro Planner	MACAZINE	Vol. 2 No.12	p. 51
Micro Planner	MACUSER	Mar. 86	p. 54
Multiuser Micros	INFOWORLD	Oct. 13, 86	p. 39
ODS Consultant	MACUSER	Apr. 86	p. 55
Pro Billing	MACWORLD	Dec. 86	p. 195
Profit Projections/Break even	A+ MAGAZINE	Mar. 86	p. 135
Project Management	INFOWORLD	Oct. 6, 86	p. 47
Puzzle 512	MACUSER	Sep. 86	p. 37
Stocks and Bonds	MACUSER	May 86	p. 75
Smart Alarms	MACWORLD	Dec. 86	p. 145
STELLA	MACWORLD	Sep. 86	p. 109
WillWriter	MACUSER	Jan. 86	p. 34
WillWriter	MACWORLD	Apr. 86	p. 140
WillWriter	MACAZINE	June 86	p. 33

Articles of Interest—
Computer Languages

BASIC	MACUSER	Jan. 86	p. 110
Applesoft to Microsoft	NIBBLE	Sep. 85	p. 52
Autolog—TeleCom	NIBBLE	Nov./Dec. 85	p. 44
BASIC: A Look at ROM	MACUSER	Jan. 86	p. 105
BASIC Multiplan Connection	NIBBLE	May/June 86	p. 46
BASIC: using CLR Libraries	MACUSER	July 86	p. 93

Articles of Interest—Computer Languages

Change Fonts and Sizes	NIBBLE	Mar./Apr 86	p. 52
Cursor Editor	NIBBLE	May/June 86	p. 26
Fractal Trees	NIBBLE	Jan./Feb. 86	p. 48
Locating Mouse Coordinates	MACWORLD	Nov. 85	p. 170
I Ching (game)	NIBBLE	Mar./Apr. 86	p. 27
Macbanker (checking account)	NIBBLE	Jan./Feb. 86	p. 9
MacBanner (sign maker)	NIBBLE	Mar./Apr. 86	p. 18
Mac File Cabinet	NIBBLE	Nov. 85	p. 9
MacGraph Program	NIBBLE	Sept. 85	p. 34
Mac Investor Program	NIBBLE	Sept. 85	p. 6
Mac Investor revisited	NIBBLE	Mar./Apr. 86	p. 66
MacNosy	A+ MAGAZINE	June 85	p. 145
MacSprint (speed reading)	NIBBLE	May/June 86	p. 9
Microsoft BASIC Tricks	NIBBLE	Nov./Dec. 85	p. 42
Microsoft BASIC Tricks	NIBBLE	Jan./Feb. 86	p. 37
Microsoft BASIC Tricks	NIBBLE	Mar./Apr. 86	p. 44
MS BASIC Program Shell	NIBBLE	May/June 86	p. 52
MS BASIC Toolbox Windows	NIBBLE	Nov. 85	p. 68
MS BASIC Windows	NIBBLE	Nov./Dec. 85	p. 69
CLR MathStatLib	MACWORLD	Dec. 86	p. 149
CLR SpeechLib	MACWORLD	Dec. 86	p. 149
CLR ToolLib	MACWORLD	Dec. 86	p. 149
True BASIC 1.2	MACWORLD	Nov. 86	p. 143
Neon	A+ MAGAZINE	March 86	p. 128
NIBBLE MacFile Cabinet	NIBBLE	Nov./Dec.85	p. 9
Pattern Designer Program	NIBBLE	Sept. 85	p. 26
Pattern Design Program in BASIC	MACWORLD	Jan. 86	p. 156
Picture Aide (puts pics in programs)	A+ MAGAZINE	Apr. 86	p. 137
Rescuing a Document	MACWORLD	Nov. 85	p. 163
Stop Watch Program	NIBBLE	Nov./Dec.85	p. 74
Zoom (Data base of Pictures)	NIBBLE	Mar./Apr. 86	p. 9

Pascal

Macintosh Pascal Primer Part I	A+ MAGAZINE	Mar. 85	p. 95
Macintosh Pascal Primer Part II	A+ MAGAZINE	Apr. 85	p. 95
Macintosh Pascal	MACWORLD	Apr. 85	p. 123
Macintosh Pascal	MACAZINE	Vol.3 No.3	p. 76
Macintosh Pascal Shell Program	NIBBLE	Nov. 85	p. 65
Pascal Programming	MACUSER	Nov. 86	p. 119

Articles of Interest—Computer Languages

✱The Feel of Pascal	MACUSER	Nov. 86	p. 119
✱The Sound of Pascal	MACUSER	Oct. 86	p. 105
Pascal	MACUSER	Jan. 86	p. 114
Pascal and the Mac Interface	MACUSER	Aug. 86	p. 105
A Taste of Pascal	MACUSER	July 86	p. 104
TML Pascal	MACUSER	June 86	p. 98

Miscellaneous Articles on Computer Languages

Aztec C	MACUSER	May 86	p. 36
To C or Not To C	A+ MAGAZINE	Apr. 85	p. 87
C-Worthy Compiler (Consulair)	A+ MAGAZINE	May 85	p. 143
C	MACUSER	Jan. 86	p. 112
C Program (Macintosh Sound)	MACUSER	Apr. 86	p. 105
C Programming	MACUSER	March 86	p. 104
C Programming	MACUSER	May 86	p. 105
C Programming	MACUSER	June 86	p. 108
✱Lightspeed C	MACWORLD	Oct. 86	p. 119
Assembly-Language Programs	A+ MAGAZINE	Mar. 85	p. 103
Assembly-Language Programming	A+ MAGAZINE	Feb. 85	p. 109
Assembly-Language Programming	A+ MAGAZINE	Feb. 85	p. 99
68000 Assembly Language III	A+ MAGAZINE	Aug. 86	p. 91
ExperLogo	INFOWORLD	June 86	p. 52
Forth	MACUSER	Jan. 86	p. 115
FORTRAN	MACWORLD	June 86	p. 110
The Macintosh Lisp Machine	MACWORLD	Mar. 86	p. 99
MacForth Level II	MACUSER	July 86	p. 36
✱MacLightning 2.0	INFOWORLD	Sep. 22, 86	p. 54
✱Mac Programming Workshop	MACWORLD	Dec. 86	p. 122
MacScheme	MACUSER	June 86	p. 36
✱MacScheme 1.11	MACWORLD	Dec. 86	p. 147
✱Mac as UNIX Workstation	INFOWORLD	Oct. 6, 86	p. 31
✱Object Programming	MACWORLD	Nov. 86	p. 119

Articles of Interest—Data Bases

Filevision

Business Filevision	MACUSER	Feb. 86	p. 46
Business Filevision	A+ MAGAZINE	Feb. 86	p. 135
Choosing a Database	MACWORLD	Aug. 86	p. 146

Articles of Interest—Data Bases

Filevision gets down to business	MACWORLD	Mar. 86	p. 94
dBase III	MACUSER	Nov. 86	p. 68
dMac III	INFOWORLD	Oct. 13, 86	p. 47
dMac III	MACUSER	Nov. 86	p. 69
Database Software	INFOWORLD	Sept. 8, 86	p. 34
File Management Software	INFOWORLD	Sept. 8, 86	p. 29
FileMaker Plus	MACUSER	Nov. 86	p. 47
Flexware	MACAZINE	May 86	p. 69
Helix 2.0	MACUSER	Mar. 86	p. 48
Helix V2.0RII	MACWORLD	June 86	p. 120
Double Helix	MACUSER	Oct. 86	p. 52
Interlace	MACUSER	Apr. 86	p. 47
Interlace	MACAZINE	May 86	p. 69
Interlace	MACWORLD	Oct. 86	p. 126
MacBase	MACUSER	July 86	p. 32
Omnis 3	POP. COMP.	Dec. 85	p. 70
Omnis 3	MACUSER	Mar. 86	p. 42
PictureBase	MACWORLD	July 86	p. 152
Record Holder	MACAZINE	May 86	p. 69
Record Holder	MACWORLD	Sept. 86	p. 136

Articles of Interest—Desk Accessories

Expanding the Apple Menu	NIBBLE	Mar./Apr. 86	p. 40
Mockchart	NIBBLE	Mar./Apr. 86	p. 42
SideKick	MACUSER	Dec. 85	p. 46
SideKick—DA	MACWORLD	Mar. 86	p. 107
SideKick	MACUSER	Dec. 85	p. 112
SideKick	NIBBLE	Mar./Apr. 86	p. 40
Teleport	NIBBLE	Mar./Apr. 86	p. 42
Quick and Dirty Utilities	NIBBLE	Mar./Apr. 86	p. 43

Articles of Interest—Education

Consortium Software	A+ MAGAZINE	May 86	p. 40
Dapona Gradebook	MACAZINE	May 86	p. 30
Dynamic Reader	MACWORLD	Oct. 86	p. 163
MicroGrade 1.0	MACWORLD	Nov. 86	p. 144
MicroTest II 1.0	MACWORLD	Nov. 86	p. 145
MacType	MACAZINE	Vol.3 No.3	p. 43
Math Flash	MACAZINE	Vol.3 No.3	p. 45
NotePRO	MACUSER	July 86	p. 36
NotePRO	MACAZINE	May 86	p. 88
SAT Score Improvement	MACUSER	May 86	p. 40
Slide Show Magician &Videoworks	MACUSER	May 86	p. 83
Spanish Grammar III	MACUSER	Apr. 86	p. 33
Speed Reader II	MACUSER	Apr. 86	p. 36
Speller Bee	MACWORLD	Apr. 86	p. 148
SpellerBee	MACAZINE	May 86	p. 90
The University Macintosh	A+ MAGAZINE	June 86	p. 111
Word Play	MACAZINE	Vol. 3 No.3	p. 43

Articles of Interest—Games

Amazon	MACUSER	Feb. 86	p. 36
The Ancient Art of War	MACWORLD	Aug. 86	p. 129
Alter Ego 1.0	MACWORLD	Dec. 86	p. 142
Archon	MACUSER	Oct. 86	p. 92
Bridge 4.0	MACUSER	Sept. 86	p. 38
Bagels 1.1	MACWORLD	Aug. 86	p. 103
Balance of Power	MACWORLD	July 86	p. 146
Billiard Parlour	MACWORLD	Aug. 1986	p. 103
Borrowed Time	MACUSER	May 86	p. 97
Brimstone	MACUSER	July 1986	p. 34
Championship Boxing	MACUSER	Mar. 86	p. 87
Championship Star League Baseball	MACWORLD	Mar. 86	p. 140
The Crimson Crown	MACUSER	Aug. 1986	p. 87
Deadline	MACUSER	Dec. 85	p. 98
Deja Vu	MACUSER	Jan. 86	p .89
Deja Vu	MACAZINE	Vol.3 No.2	p. 38
Enchanted Scepters	MACUSER	May 86	p. 37
Enchanted Scepters	MACWORLD	June 86	p. 145
Flight Simulator	MACWORLD	Aug. 86	p. 129

Articles of Interest—Games

Flight Simulator	A+ MAGAZINE	Aug. 86	p. 72
Fokker Triplane	MACUSER	Feb. 86	p. 93
Felony	MACUSER	Dec. 85	p. 98
Flight Simulator	MACUSER	Nov. 86	p. 101
Gato	MACWORLD	Dec. 85	p. 125
Grid Wars	MACUSER	Apr. 86	p. 33
Grid Wars	MACAZINE	Vol. 2 No.12	p. 35
Grand Slam	MACUSER	Nov. 86	p. 51
Grand Slam Tennis	MACWORLD	Dec. 86	p. 141
Harrier Strike Mission	MACUSER	Feb. 86	p. 93
Harrier Strike Mission	MACUSER	Jan. 86	p. 34
Infidel	MACUSER	Feb. 86	p. 32
KidsTime	MACWORLD	Sept. 86	p. 153
KidsTime	MACUSER	Oct. 86	p. 33
MacChallenger	MACUSER	Feb. 86	p. 93
MacGolf	MACWORLD	Aug. 86	p. 144
MacGolf	MACAZINE	May 86	p. 63
MacGolf	MACUSER	May 86	p. 94
MazeWars+	MACUSER	Nov. 86	p. 51
Mind Over Mac	MACUSER	Feb. 86	p. 32
MindShadow	MACWORLD	June 86	p. 160
The Mist	MACUSER	June 86	p. 97
Murder By the Dozen	MACUSER	Dec. 85	p. 98
One on One	MACWORLD	Sep. 86	p. 178
Orbiter	MACUSER	July 86	p. 34
Orbiter	MACUSER	Nov. 86	p. 103
Orbiter	MACWORLD	Sep. 86	p. 164
Perplexx	MACUSER	Oct. 86	p. 102
Pinball Construction Set	MACUSER	Jan. 86	p. 36
Ractor	MACWORLD	June 86	p. 148
Rogue	MACUSER	Feb. 86	p. 86
Sargon III vs. Psion	MACUSER	Mar. 86	p. 81
Shanghai	MACUSER	Nov. 86	p. 110
Shanghai	MACUSER	Oct. 86	p. 38
Skyfox	MACUSER	Oct. 86	p. 33
Strategic Conquest	MACUSER	Dec. 85	p. 34
Suspect	MACUSER	Dec. 85	p. 98

Through the Looking Glass	MACUSER	Jan. 86	p. 33
✱The Toy Shop	MACWORLD	Dec. 86	p. 152
✱The Toy Shop	MACUSER	Nov. 86	p. 47
Transylvania	MACUSER	Aug. 86	p. 87
Tycoon	NIBBLE	Jan./Feb. 86	p. 46
Ultima III	MACAZINE	Vol.2 No.12	p. 92
✱Uninvited	MACWORLD	Sep. 86	p. 158
✱Uninvited	MACUSER	Sep. 86	p. 103
Universe II	MACUSER	Jan. 86	p. 95
✱Winter Games 1.0	MACWORLD	Dec. 86	p. 150
Witness	MACUSER	Dec. 85	p. 98
✱Word Challenge II	MACUSER	Oct. 86	p. 100
✱WordPlay	MACUSER	Oct. 86	p. 98
✱World Builder 1.0	MACWORLD	Nov. 86	p. 138
Wizardry	MACWORLD	Apr. 86	p. 138
Wizardry	MACUSER	June 86	p. 94
Wizardry	MACAZINE	May 86	p. 62
Zork	MACUSER	Dec. 85	p. 97
✱The Game Hall of Fame	MACWORLD	Dec. 86	p. 119

Articles of Interest—Graphic Design

Cheap Paint	MACAZINE	Vol.3 No.3	p. 22
ClickArt Effects	MACUSER	Apr. 86	p. 35
✱Clip Art Collections	MACWORLD	Dec. 86	p. 133
✱ColorChart	MACWORLD	Oct. 86	p. 152
Color Macintosh	MACUSER	May 86	p. 61
ColorMate	MACWORLD	July 86	p. 117
✱ColorPress 1.1	MACWORLD	Dec. 86	p. 141
ColorPrint	MACWORLD	July 86	p. 95
✱ComicWorks	MACUSER	Nov. 86	p. 47
✱ComicWorks	MACWORLD	Sept. 86	p. 133
Comic Works	MACWORLD	Aug. 86	p. 59

Articles of Interest—Graphics

Cricket Graph	MACWORLD	July 86	p. 95
Cricket Graph	MACUSER	June 86	p. 37
✹Cricket Graph	MACUSER	Nov. 86	p. 83
Cutting, Pasting, Using Graphics	MACWORLD	Jan. 86	p. 68
✹DrawArt	MACUSER	Oct. 86	p. 36
Easy3D	MACUSER	May 86	p. 69
✹EZ-Draft	MACWORLD	Nov. 86	p. 116
✹Full Paint	MACWORLD	Sept. 86	p. 142
✹Full Paint	MACUSER	Oct. 86	p. 79
Full Paint	MACUSER	June 86	p. 36
✹Glue 1.04	MACWORLD	Dec. 86	p. 144
Graphing in 3D	MACAZINE	Vol.3 No.3	p. 78
✹The Graphic Mac	MACWORLD	Nov. 86	p. 101
✹The Graphics Magician	MACWORLD	Oct. 86	p. 154
✹Gridmaker	MACUSER	Nov. 86	p. 53
Home Design	MACUSER	June 86	p. 38
Icon Collector V 1.1	MACAZINE	July 86	p. 43
Mac 3D	MACWORLD	June 86	p. 96
✹Mac-3D	INFOWORLD	Oct. 10, 86	p. 9
Mac-A-Mug	MACUSER	July 86	p. 75
✹MacBillboard 4.01	MACWORLD	Nov. 86	p. 140
MacDraft	MACUSER	Feb. 86	p. 81
MacDraft	MACUSER	Dec. 85	p. 36
MacDraft	MACAZINE	Vol.3 No.1	p. 76
MacDraft	MACWORLD	Jan. 86	p. 92
MacDraft vs. MacDraw	MACAZINE	Vol.3 No.1	p. 97
✹MacDraft	MACWORLD	Nov. 86	p. 114
MacIntizer	MACUSER	Jan. 86	p. 86
✹MacMap	MACWORLD	Nov. 86	p. 146
MacModel	MACWORLD	June 6	p. 99
MacPaint V 1.0	MACAZINE	July 86	p. 42
✹MacPerspective 2.0	MACWORLD	Nov. 86	p. 142
Macromind Utilities	MACUSER	Feb. 86	p. 36
MacSpace	MACWORLD	June 86	p. 98
MacSpin	MACWORLD	Aug. 86	p. 97
MacTablet	MACUSER	Jan. 86	p. 84
Mac The Ripper	MACUSER	May 86	p. 40
✹MacVideo Processor	MACWORLD	Dec. 86	p. 146
Mac Viz	MACAZINE	May 86	p. 81
Magic Slate	MACUSER	Feb. 86	p. 32
✹MGMStation	MACWORLD	Nov. 86	p. 117
Paste-Ease V1	MACAZINE	Vol.3 No.3	p. 19

⌘Phoenix 3D	MACUSER	Oct. 86	p. 38
PictureBase	MACUSER	June 86	p. 38
PictureBase	MACAZINE	July 86	p. 54
⌘PictureBase	MACUSER	Sept. 86	p. 60
Picture Perfect Presentation	MACUSER	July 86	p. 60
⌘Plotter Drivers	MACWORLD	Oct. 86	p. 96
PosterMaker	MACAZINE	July 86	p. 40
⌘PosterMaker	MACUSER	Sept. 86	p. 42
⌘PosterMaker 1.0	MACWORLD	Nov. 86	p. 140
⌘Preparing Charts	MACUSER	Sept. 86	p. S5
⌘Presentation Graphics	INFOWORLD	Sept. 22, 86	p. 35
Silicon Press	MACUSER	Aug. 86	p. 75
Slide Show Magician	MACWORLD	Aug. 86	p. 132
Slide Show Magician	A+ MAGAZINE	Jan. 86	p. 94
⌘Storyboarder	MACUSER	Nov. 86	p. 89
⌘SuperPaint 1.0	MACWORLD	Nov. 86	p. 133
VideoWorks	MACWORLD	Aug. 86	p. 108
Working with Clipboard, Scrapbook, Notepad	MACUSER	Mar. 86	p. 91

Articles of Interest—Hardware

Hard Disk Decisions	MACWORLD	June 86	p. 103
Making the Most of an HD	MACUSER	July 86	p. 44
HD20 and the HyperDrive 20	A+ MAGAZINE	Feb. 86	p. 125
HyperDrive	MACUSER	Feb. 86	p. 63
Mac 10 (hard disk)	MACAZINE	Vol.3 No.1	p. 65
Hard Disk 20	MACWORLD	Jan. 86	p. 77
⌘Double-Sided Drives	MACWORLD	Nov. 86	p. 106
⌘Enter the SCSI Drive	MACWORLD	Sept. 86	p. 98
⌘Hard Disk Drives	MACUSER	Sept. 86	p. 52
⌘AST-2000 Hard Disk	MACWORLD	Dec. 86	p. 143
MacBottom	MACAZINE	Vol.3 No.1	p. 24

Articles of Interest—Hardware

Optical Disks	MACWORLD	Feb. 86	p. 100
MacCharlie	POP. COMP.	Dec. 85	p. 65
MacCharlie	MACUSER	Dec. 85	p. 60
MacCharlie	MACWORLD	Feb. 86	p. 120
MacCharlie (file transfer)	A+ MAGAZINE	Jan. 86	p. 129
MacIntizer	MACUSER	Jan. 86	p. 86
Abaton (file transfer)	A+ MAGAZINE	Jan. 86	p. 129
Choosing Apple Hardware	MACUSER	May 86	p. 42
ImageWriter II	MACWORLD	Jan. 86	p. 86
LaserWriter Revealed	MACUSER	Feb. 86	p. 55
Macintosh Plus	MACWORLD	Feb. 86	p. 57
Macintosh Plus	A+ MAGAZINE	Mar. 86	p. 22
Macintosh Plus	MACUSER	Mar. 86	p. 39
Making the Hardware Connection	A+ MAGAZINE	Jan. 86	p. 30
MacTablet	MACUSER	Jan. 86	p. 83
Memory Upgrades	MACWORLD	Feb. 86	p. 114
5 Memory Upgrades	MACWORLD	Feb. 86	p. 115
Networking Hardware and Software	MACAZINE	Vol.3 No.3	p. 87
Penmouse +	MACWORLD	Aug. 86	p. 146
Printing in Color	MACWORLD	Feb. 86	p. 114
ProModem 1200	MACWORLD	Apr. 86	p. 129
RAMSTAK (memory for the XL)	MACAZINE	Vol.3 No.3	p. 42
MacInker	MACWORLD	Sept. 86	p. 162
Radius Full-Page Display	MACUSER	Nov. 86	p. 57
Software Strip Reader	MACUSER	Apr. 86	p. 38
Softstrip	MACWORLD	Oct. 86	p. 164
HardwareCauzin Strip Reader	MACAZINE	Aug. 86	p. 20

Articles of Interest—Miscellaneous

Astronomy V2.8	MACUSER	Apr. 86	p. 79
Calculator Construction Set	MACWORLD	Aug. 86	p. 134
Calliope	MACUSER	Sept. 86	p. 40
Design Scope	MACUSER	Nov. 86	p. 43
Developing a Megabyte Strategy	MACWORLD	Apr. 86	p. 80
DietMac	MACUSER	May 86	p. 38

Dinner at Eight	MACUSER	Jan. 86	p. 32
The Finder	MACUSER	Oct. 86	p. 111
Front Desk	MACUSER	Dec. 85	p. 36
Front Desk	INFOWORLD	July 21, 86	p. 34
Hippo Almanac	MACUSER	Jan. 86	p. 33
HomePak	MACUSER	Sept. 86	p. 40
How Disks Work	MACWORLD	Dec. 86	p. 171
Insights into ImageWriter	MACWORLD	Nov. 86	p. 183
The LABtop Macintosh	MACWORLD	Oct. 86	p. 137
LaserWriter Capabilities	INFOWORLD	Aug. 25, 86	p. 41
LogiMac (digital logic design)	MACAZINE	Vol.3 No.1	p. 96
Mac Software: The Market	INFOWORLD	June. 30, 86	p. 25
Macintosh vs. IBM Software	INFOWORLD	June 30, 86	p. 26
Mac+II	MACWORLD	Oct. 86	p. 161
MacPlus System	MACWORLD	Apr. 86	p. 86
MacSpin	MACUSER	June 86	p. 40
Macstronomy	MACUSER	Apr. 86	p. 80
Quick Circuit 3.0	MACWORLD	Dec. 86	p. 150
SCSI Macport	A+ MAGAZINE	Feb. 86	p. 157
Smart Eyes	MACWORLD	Oct. 86	p. 163
Speeding Up the Mac	MACWORLD	Aug. 86	p. 88
Starting a New Business	MACUSER	Dec. 85	p. 36
StatView	MACUSER	Dec. 85	p. 38
StatView	MACUSER	May 86	p. 46
StatView 512+ 1.0	MACWORLD	Dec. 86	p. 138
Stat 80	MACWORLD	Nov. 86	p. 109
StatWorks	MACUSER	May 86	p. 46
Stella	MACAZINE	May 86	p. 53
Time Base (Calendar)	A+ MAGAZINE	Nov. 85	p. 164
Tellstar	MACAZINE	Vol.3 No.1	p. 62
Tellstar I & II	MACUSER	Apr. 86	p. 82
Virtually Free Software	NIBBLE	May/June 86	p. 58
X-10 Powerhouse	MACWORLD	Nov. 86	p. 137

Articles of Interest—Music

Enhancing a Music Program	MACAZINE	June 86	p. 28
Concertware	MACAZINE	Mar. 85	p. 29
✱ConcertWare+MIDI	MACWORLD	Dec. 86	p. 112
Deluxe Music Construction Set	MACUSER	Mar. 86	p. 75
Deluxe Music Construction Set	MACUSER	Feb. 86	p. 34
Deluxe Music Construction Set	MACAZINE	Vol.2 No.12	p. 84
✱Deluxe Music Construction Set	MACWORLD	Dec. 86	p. 112
✱High Score	MACWORLD	Dec. 86	p. 117
✱MegaTrack XL	MACWORLD	Sept. 86	p. 172
Midi for the Macintosh	A+ MAGAZINE	Feb. 86	p. 117
✱MIDI Life Crisis	MACWORLD	Sept. 86	p. 27
✱MIDIMAC Sequencer	MACWORLD	Dec. 86	p. 114
✱Music on the Mac+	MACAZINE	Aug. 86	p. 24
Music on the Mac	MACAZINE	Vol. 3 No. 3	p. 56
✱Music Processing	MACWORLD	Dec. 86	p. 109
Music Software	MACWORLD	Feb. 86	p. 92
✱Performer	MACWORLD	Dec. 86	p. 115
✱Performer 1.22	MACWORLD	Dec. 86	p. 140
✱Professional Composer	MACWORLD	Dec. 86	p. 115
Professional Composer V2	MACWORLD	Feb. 86	p. 109
Professional Composer V2	MACAZINE	Vol.2 No.12	p. 85
Total Music	MACUSER	Jan. 86	p. 34

Articles of Interest—Spreadsheets

Common Spreadsheet Mistakes	MACUSER	Aug. 86	p. 47
Crunch	MACUSER	Jan. 86	p. 61
Ensemble	MACUSER	July 86	p. 50
Excel	MACUSER	Dec. 85	p. 41
Excel Macros	MACWORLD	Dec. 85	p. 69
MACROMAKER New for Excel	MACUSER	Dec. 85	p. 40
Excel Functions	A+ MAGAZINE	Mar. 86	p. 141
Excel Databases	MACWORLD	June 86	p. 116
✱Four Secrets of Excel	MACWORLD	Sept. 86	p. 121
Executive Office	MACWORLD	July 86	p. 108

⌘Jazz 1A	MACWORLD	Dec. 86	p. 103
⌘Jazz 1A	INFOWORLD	Sept. 8, 86	p. 41
Jazz	MACWORLD	Jan. 86	p. 96
Jazz	MACWORLD	Mar. 86	p. 23
⌘MacCalc 1.0	MACWORLD	Dec. 86	p. 148
⌘MacCalc	MACUSER	Sep. 86	p. 38
⌘Microsoft Works	INFOWORLD	Sep. 22, 86	p. 48
⌘Microsoft Works	MACUSER	Nov. 86	p. 61
⌘Microsoft Works 1.0	MACWORLD	Dec. 86	p. 103
⌘Microsoft Works	MACWORLD	Oct. 86	p. 113
⌘Microsoft Works	MACUSER	Oct. 86	p. 42
Quartet	MACUSER	July 86	p. 50
Spreadsheets for the Mac	POP. COMP.	Aug. 85	p. 12

Articles of Interest—Telecommunications

AppleTalk	MACUSER	Mar. 86	p. 68
⌘GEnie (on line service)	MACWORLD	Nov. 86	p. 141
Going On-Line with Excel	MACWORLD	July 86	p. 126
⌘Integrating Mac and Vax	INFOWORLD	Oct. 26, 86	p. 1
InTouch	MACWORLD	Dec. 85	p. 90
InTouch 2.05	NIBBLE	May/June 86	p. 41
⌘Maccessories Modem	MACWORLD	Sept. 86	p. 176
MacCommunications	A+ MAGAZINE	Jan. 86	p. 137
Macintosh Networks	MACWORLD	June 86	p. 127
MacTerminal	MACUSER	Dec. 85	p. 85
⌘MAUG	MACUSER	Oct. 86	p. 87
MicroPhone 1.0	NIBBLE	May/June 86	p. 40
Microphone	MACUSER	July 86	p. 69
Microphone	MACWORLD	July 86	p. 133
Mini & Mainframe Connections	MACWORLD	July 1986	p. 121
Modems	MACUSER	Feb. 86	p. 74
⌘Modems and Communications	INFOWORLD	Sept. 15, 86	p. 25
⌘Networks	MACWORLD	Oct. 86	p. 104
The PBX Connection	MACWORLD	July 86	p. 123

Articles of Interest—Telecommunications

Red Ryder	MACWORLD	Dec. 85	p. 92
Red Ryder 6.0	MACAZINE	Vol.3 No.1	p. 91
Red Ryder 6.2c	MACAZINE	Vol.2 No.12	p. 21
Red Ryder 7.0	NIBBLE	May/June 86	p. 43
Red Ryder	MACUSER	July 86	p. 71
Smartcom	MACUSER	July 86	p. 72
Smartcom II	MACAZINE	Vol.2 No.12	p. 22
Smartcom II	MACWORLD	Dec. 85	p. 88
Smartcom II 2.1D	NIBBLE	May/June 86	p. 39
Telecommunications	MACUSER	Nov. 86	p. 97
Turbo Download 3.1	MACWORLD	Nov. 86	p. 145
Unix to Mac Networks	MACWORLD	June 86	p. 129

Articles of Interest—Utilities

AutoMac 1.5	MACAZINE	Aug. 86	p. 60
Copy II Mac	MACUSER	June 86	p. 49
Catalog Keeper	MACUSER	Aug. 86	p. 92
DA Installer	A+ MAGAZINE	Aug. 86	p. 99
Disk Info	MACUSER	Aug. 86	p. 96
DiskInfo	MACWORLD	Nov. 86	p. 126
Disk Librarian	MACWORLD	Nov. 86	p. 127
Disk Ranger	MACWORLD	Nov. 86	p. 128
Disk Ranger	MACUSER	Aug. 86	p. 94
Document Compare	MACUSER	May 86	p. 37
Extras	MACUSER	Aug. 86	p. 94
Favorite Desk Accessories	MACAZINE	Aug. 86	p. 34
FEDIT Plus	MACUSER	Sept. 86	p. 40
F Key Installer	MACWORLD	Aug. 86	p. 102
Flashback	MACUSER	Nov. 86	p. 53
Flashback	MACWORLD	Nov. 86	p. 128
FlashFinder V2.1	MACAZINE	Vol.3 No.3	p. 20
HD Backup	MACWORLD	Nov. 86	p. 129

HFS Backup	MACWORLD	Nov. 86	p. 129
HFS Backup	MACUSER	Aug. 86	p. 94
HFS Locator	MACWORLD	Nov. 86	p. 125
HFS Locator Plus	MACUSER	Sept. 86	p. 42
Hippolock	MACAZINE	Aug. 86	p. 44
The HouseKeeper	MACUSER	Dec. 85	p. 35
Icon Switcher 1.2	A+ MAGAZINE	Aug. 86	p. 101
Keyboard Conversion	MACWORLD	Oct. 86	p. 131
MacBuffer 2.2	MACWORLD	Nov. 86	p. 135
MacDisk Catalog	MACUSER	Dec. 85	p. 35
MacExpress	MACUSER	Feb. 86	p. 33
Macintosh Resources	MACUSER	Sept. 86	p. 108
MacKeymeleon	MACWORLD	Sept. 86	p. 159
MacMemory Disk	MACUSER	June 86	p. 36
MacPlot	MACUSER	Dec. 85	p. 34
MacQWERTY	MACUSER	July 86	p. 33
Mac spoolers for LaserWriter	INFOWORLD	Oct. 6, 86	p. 16
MacZap	MACUSER	Jan. 86	p. 32
MDCII	MACWORLD	Nov. 86	p. 127
MenuEdit 1.0	A+ MAGAZINE	Aug. 86	p. 100
Mock Printer	MACUSER	Aug. 86	p. 98
MockTerminal	MACUSER	Aug. 86	p. 98
MockWrite	MACUSER	Aug. 86	p. 98
Modify Your System	MACUSER	Sept. 86	p. 121
My Disk Labeler	MACWORLD	Aug. 86	p. 135
My Disk Labeler V1	MACAZINE	Vol.3 No.3	p. 19
N'Cryptor	MACAZINE	Aug. 86	p. 44
Packer	MACUSER	June 86	p. 36
Packer	MACWORLD	Aug. 86	p. 142
PaintMover	MACWORLD	Aug. 86	p. 101
P/C Privacy	MACAZINE	Aug. 86	p. 43
Printer Server/Buffer	MACWORLD	Nov. 86	p. 135
Print Spoolers	MACUSER	Aug. 86	p. 58
Q + D Filer	MACUSER	Aug. 86	p. 94
RAM Start	MACWORLD	Aug. 86	p. 100
REdit	A+ MAGAZINE	Aug. 86	p. 100
Smart Alarms	MACUSER	Oct. 86	p. 36
Sort Menu	A+ MAGAZINE	Aug. 86	p. 100
Switcher	MACUSER	Aug. 86	p. 98
Switcher	MACUSER	Dec. 85	p. 120
Switcher	MACUSER	June 86	p. 48

Articles of Interest—Utilities

Taming the Hard Disk	MACWORLD	Nov. 86	p. 124
Tempo	MACAZINE	June 86	p. 89
Tempo	MACUSER	July 86	p. 38
Timebase	MACWORLD	June 86	p. 150
TopDesk	MACWORLD	June 86	p. 137
TopDesk	MACUSER	May 86	p. 36
TypeNow	MACUSER	Jan. 86	p. 36
Transfer 2.0	MACWORLD	Aug. 86	p. 100
Waystation 1.9B	MACWORLD	Aug. 86	p. 100

Articles of Interest—Word Processing

ABC's of Desktop Publishing	MACUSER	June 86	Suppl.
Acta	MACWORLD	Oct. 86	p. 150
Acta (desk accessory outliner)	MACUSER	July 86	p. 32
Desktop Publishing Directory	MACUSER	Sept. 86	p. S30
Desktop Publishing Tips	MACUSER	Sept. 86	p. S17
Document Modeler	MACWORLD	Aug. 86	p. 104
EGWORD (Japanese WP)	MACUSER	Apr. 86	p. 60
Font Librarian	A+ MAGAZINE	Aug. 86	p. 99
Fontographer	MACWORLD	July 86	p. 142
Fontographer	MACAZINE	May 86	p. 45
Haba Word 1.0	MACWORLD	Dec. 86	p. 101
Hayden Speller	MACWORLD	Jan. 86	p. 131
Hayden Speller	MACAZINE	June 86	p. 81
Hayden Speller	MACWORLD	July 86	p. 105
Hayden Speller 1.2	MACUSER	Oct. 86	p. 68
Just Text	MACUSER	Dec. 85	p. 76
Laser Quill 1.2	MACWORLD	Nov. 86	p. 130
Laser Quill 1.2	MACWORLD	Dec. 86	p. 101
LaserWorks	MACUSER	Sept. 86	p. 37
LaserWorks 1.1	MACWORLD	Nov. 86	p. 136
Liberty Spell Checker 2.0	MACUSER	Oct. 86	p. 69
Liberty Spell-Checker	MACWORLD	July 86	p. 103
The Linotronic Connection	MACUSER	Sept. 86	p. S20

MacAuthor	MACAZINE	Vol.3 No.1	p. 60
MacGas	MACWORLD	July 86	p. 105
MacGas 1.0	MACUSER	Oct. 86	p. 69
MacIndexer	MACUSER	July 86	p. 33
MacIndexer	MACWORLD	June 86	p. 138
MacLightning	MACWORLD	July 86	p. 103
MacLightning 2.0	MACUSER	Oct. 86	p. 70
MacPublisher II	INFOWORLD	Oct. 13, 86	p. 54
MacPublisher II	MACUSER	Sept. 86	p. S22
MacSpec	MACUSER	Oct. 86	p. 34
Mac•Spell•Right	MACWORLD	July 86	p. 106
Mac•Spell•Right	MACAZINE	June 86	p. 83
Mac Spell Right 3.3	MACUSER	Oct. 86	p. 71
MacSpell+	MACWORLD	Mar. 86	p. 130
MacSpell+	MACAZINE	June 86	p. 82
MacSpell+	MACWORLD	July 86	p. 106
MacSpell+ 1.1	MACUSER	Oct. 86	p. 70
MacWrite 4.5	MACWORLD	Dec. 86	p. 100
MaxThink	MACWORLD	Sept. 86	p. 148
MaxThink	MACUSER	Oct. 86	p. 34
Microsoft Word Features	MACWORLD	Apr. 86	p. 112
Microsoft Word 1.05	MACWORLD	Dec. 86	p. 100
Microsoft Word 3.0	MACWORLD	Dec. 86	p. 101
MindWrite 1.0	MACWORLD	Dec. 86	p. 102
More	MACWORLD	Sept. 86	p. 112
More	MACUSER	Sept. 86	p. 44
More (outliner)	INFOWORLD	Sept. 15, 86	p. 39
Outline Processors	MACWORLD	Aug. 86	p. 61
Using PageMaker	MACUSER	Nov. 86	p. 74
PageMaker Shortcuts	MACWORLD	Sept. 86	p. 117
PageMaker	MACWORLD	Mar. 86	p. 11
PageMaker	MACUSER	June 86	Suppl.
Bringing PageMaker to Life	MACUSER	Nov. 86	p. 74
Power Word Processing	MACUSER	Sept. 86	p. S12
ReadySetGo	MACUSER	Apr. 86	p. 69
The Right Word 3.0	MACUSER	Oct. 86	p. 72
RightWord	MACAZINE	June 86	p. 83
RightWord	MACWORLD	July 86	p. 106
Professional Bibliographic System	MACWORLD	Jan. 86	p. 45
Professional Bibliographic System	MACUSER	Feb. 86	p. 33
Professional Bibliographic System	MACAZINE	Vol.3 No.3	p. 34

Articles of Interest—Word Processing

The Scoop on Newsletters	MACUSER	June 86	Suppl.
Secrets of Word	MACUSER	June 86	Suppl.
⌘Spelling Champion 2.0	MACUSER	Oct. 86	p. 72
⌘Spellswell 1.2c	MACUSER	Oct. 86	p. 73
SpellSwell	MACUSER	May 86	p. 38
SpellSwell	MACWORLD	July 86	p. 106
SpellSwell V1.2c	INFOWORLD	July 14, 86	p. 53
ThinkTank	MACUSER	Jan. 86	p. 52
ThinkTank 512	MACWORLD	Apr. 86	p. 106
⌘Word Handler	MACWORLD	Oct. 86	p. 143
⌘Word Handler	MACUSER	Oct. 86	p. 34
⌘Word Handler 1.6	MACWORLD	Dec. 86	p. 102
⌘Word Processing Software	INFOWORLD	Sept. 29, 86	p. 37
⌘Write Now	INFOWORLD	Oct. 10, 86	p. 44
⌘WriteNow 1.0	MACWORLD	Dec. 86	p. 103

Word Processing—Miscellaneous

The Business of Desktop Publish.	MACUSER	June 86	Suppl.
Creating Resumes	MACUSER	June 86	p. 91
Desktop Publishing Directory	MACUSER	June 86	Suppl.
You're My Type	MACUSER	June 86	Suppl.
⌘Nine Spelling Checkers	MACUSER	Oct. 86	p. 68

Word Processing—Fonts

ClickArt Letters	A+ MAGAZINE	Jan. 86	p. 146
Decowriter	A+ MAGAZINE	Jan. 86	p. 146
Font Agenix	A+ MAGAZINE	Jan. 86	p. 147
FONTastic	MACWORLD	Feb. 86	p. 126
FONTastic	MACAZINE	Vol.3 No.3	p. 55
Font Explorer	MACAZINE	Vol.3 No.3	p. 20
Fluent Fonts	A+ MAGAZINE	Jan. 86	p. 147
Headline Graphics	A+ MAGAZINE	Jan. 86	p. 147
Kensington Professional Fonts	A+ MAGAZINE	Jan. 86	p. 148
Mac The Knife: World Class Fonts	A+ MAGAZINE	Jan. 86	p. 148
More Fun with FONTastic	MACAZINE	Vol.3 No.3	p. 58
Sci Fonts	MACAZINE	Vol.3 No.3	p. 20
Set Default Font	A+ MAGAZINE	Aug. 86	p. 104
Superfonts	A+ MAGAZINE	Jan. 86	p. 148
Tech: Math Science Fonts	A+ MAGAZINE	Jan. 86	p. 149
UltraFonts Edition 2	A+ MAGAZINE	Jan. 86	p. 149

Additional Macintosh Resources—*MacInTouch* Newsletter

The *MacInTouch* newsletter is a journal for professionals using the Macintosh computer in business, education, and research. The publishers have given us permission to list their newsletter index along with our own resources. We feel that this addition to the *Wheels for the Mind* **Resources** section will make it even more valuable to our readers.

MacInTouch is published monthly by Ford-LePage, Inc., and it carries no advertising. Subscriptions are $48/year in the U.S.A. and Canada, delivered by third-class mail—delivery by first-class mail is an additional $9/year. Foreign subscriptions are $85/year (airmail).

Single issues and back issues are available for $5 each.

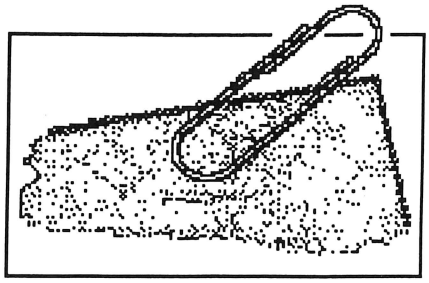
Ford-LePage, Inc.
P.O. Box 786
Framingham, MA 01701
(617) 661-8609
(617) 881-6554
Delphi: MACINTOUCH
CompuServe: [75056,1225] or [75016,350]

Subject	Article	Area	Page	Issue	V.	#
Business	Decision Support and the Macintosh	software	10	Jan-86	1	7
Cache	New MacBooster	news	29	Dec-85	1	6
Cache	Macbooster vs. Turbocharger	software	17	Jul-85	1	2
CAD	MacDraft	software	08	Oct-85	1	4
CAD	Phoenix 3D Arrives	software	11	Aug-86	2	3
Copy prot.	Copy protection: A User's View	analysis	08	Jul-85	1	2
Copy prot.	Copy Protection Methods: A Survey	analysis	12	Jul-85	1	2
Database	Choosing Database Managers	analysis	11	Nov-85	1	5
Database	Using Omnis 3	analysis	20	Nov-85	1	5
Database	Omnis 3: taking care of business?	software	01	Jul-85	1	2
Database	OverVUE	software	16	Nov-85	1	5
Database	Helix 2.0	software	05	Jan-86	1	7
Database	Record Holder	software	32	Feb-86	1	8
Database	Double Helix: double opportunities	software	09	Apr-86	1	10
Desktop	Desktop Pub. I—The Industry	analysis	01	May-86	1	11
Desktop	Desktop Pub. II—PostScript/HW	analysis	06	May-86	1	11
Desktop	Desktop Pub. III—Pagination/SW	analysis	18	Jun-86	2	1
Desktop	Microtek Scanner	news	27	Dec-85	1	6
Desktop	PageMaker: A Cut Above the Rest	software	01	Aug-85	1	3
Desktop	ReadySetGo, Version 2.0	software	05	Nov-85	1	5
File system	Macintosh Floppy Disk Organization	tech	30	Jun-85	1	1
File system	Macintosh File Organization	tech	36	Jul-85	1	2
Graphics	VideoWorks	software	23	Aug-85	1	3

Subject	Article	Area	Page	Issue	V.	#
Hard disk	Hard Disks: an overview	analysis	14	Dec-85	1	6
Hard disk	A Look at Macbottom	hardware	06	Aug-85	1	3
Hard disk	Paradise Macintosh 10	hardware	07	Nov-85	1	5
Hard disk	HyperDrive 20	hardware	17	Dec-85	1	6
Hard disk	MicahDrive AT	hardware	11	May-86	1	11
Hard disk	DataFrame 20	hardware	06	Jun-86	2	1
Hard disk	Bernoulli Box 10+10: Solid Storage	hardware	01	Aug-86	2	3
Hard disk	Mirror MagNet 20x and MacServe	hardware	07	Aug-86	2	3
Hard disk	Corvus OmniDrive/OmniTalk	Mixed	14	Jan-86	1	7
Industry	ThinkTank "upgrade?"	analysis	31	Jul-85	1	2
Industry	Apple Talk	analysis	26	Aug-85	1	3
Industry	HyperDrive 2000-HyperNet: GCC	analysis	01	Feb-86	1	8
Industry	A Macintosh hacker meets Atari's ST	analysis	17	Mar-86	1	9
Industry	Hertzfeld Conference Tidbits	news	42	Aug-85	1	3
Industry	Boston Macworld Expo	news	01	Oct-85	1	4
Industry	COMDEX Update	news	22	Dec-85	1	6
Industry	January Announcements Summary	news	04	Feb-86	1	8
Industry	Notes from San Francisco	news	01	Mar-86	1	9
Industry	Gates Conference Tidbits	Telecomm	35	Dec-85	1	6
Macintosh	The '86 Macintosh	analysis	01	Jan-86	1	7
Macintosh	Macintosh Plus What?	analysis	14	Mar-86	1	9
Macintosh	Macintosh Components/Compatibility	analysis	20	May-86	1	11
Macintosh	HFS vs. MFS	analysis	18	Aug-86	2	3
Macintosh	Multi Macintosh: Multi-Tasker	news	11	Feb-86	1	8
Macintosh	Macintosh Plus Cabling Solutions	news	16	Mar-86	1	9
Mixed	Inexpensive Macintosh Utilities	news	49	Aug-85	1	3
Music	MIDI Plays on the Macintosh	Mixed	19	Aug-85	1	3
Music	Macintosh Musicware	Mixed	18	Oct-85	1	4
Music	MIDI Notes	Mixed	30	Nov-85	1	5
Music	Macintosh Music Notes: Digitizers	Mixed	18	Jan-86	1	7
Music	MegaTrack Makes the Grade	Mixed	13	Feb-86	1	8
Music	Musicware: upgrades and updates	Mixed	12	Apr-86	1	10
Music	MIDI Goes One Step Further	Mixed	24	Jun-86	2	1
Music	The Sound of Things to Come	Mixed	14	Aug-86	2	3
Music	Performer	software	09	Dec-85	1	6
Network	Macintosh Inter-Networking	analysis	18	Mar-86	1	9
Network	What's New in Networking	news	09	Feb-86	1	8
Network	XL/Serve	software	16	Aug-85	1	3
Perform.	Macintosh Performance Matters	analysis	11	Jun-85	1	1
Perform.	MacInTouch Disk Benchmarks	analysis	10	Aug-85	1	3
Perform.	MacBooster Preview	news	13	Jun-85	1	1
Prog.	Fast Application Development	analysis	04	Dec-85	1	6
Prog.	Megamax Version 2.1	news	27	Oct-85	1	4
Prog.	Macintosh C	software	09	Nov-85	1	5
Prog.	Real Pascal for the Macintosh	software	01	Dec-85	1	6
Prog.	MacExpress	software	06	Dec-85	1	6
Prog.	ExperLisp	software	25	Jan-86	1	7
Prog.	TMON—The MONitor	software	28	Jan-86	1	7
Prog.	Neon	software	16	Apr-86	1	10
Prog.	LightspeedC	software	12	Jun-86	2	1
RAM	External RAM Disks	hardware	13	May-86	1	11

Subject	Article	Area	Page	Issue	V.	#
RAM	External RAM Disk Update	news	19	Aug-86	2	3
Resources	Owned Resources Explained	tech	31	Oct-85	1	4
ROM	The Macintosh ROM	tech	55	Aug-85	1	3
ROM	ROM Runners Beware	tech	58	Aug-85	1	3
Service	Maintenance Contracts	analysis	28	Nov-85	1	5
Service	Honeywell Service	news	30	Dec-85	1	6
Spreadsheet	First Measure of Jazz	software	01	Jun-85	1	1
Spreadsheet	Excel: Call Me Mister Spreadsheet	software	01	Nov-85	1	5
Statistics	Statistics Packages	software	01	Jul-86	2	2
Telecomm	Telecommunications Software	analysis	37	Aug-85	1	3
Telecomm	Introduction to BIX	how-to	25	Feb-86	1	8
Telecomm	File Transfer on Delphi	how-to	24	Apr-86	1	10
Telecomm	GEnie	how-to	14	Jul-86	2	2
Telecomm	About Delphi...	news	21	Jun-85	1	1
Telecomm	MacNosy Comes to Delphi	news	53	Aug-85	1	3
Telecomm	Telecommunications	news	33	Oct-85	1	4
Telecomm	AppleLink: the next generation	news	21	Feb-86	1	8
Telecomm	The Performing Arts Network	news	20	Mar-86	1	9
Telecomm	New Delphi Organization	news	23	Jul-86	2	2
Telecomm	Red Ryder 7.0	software	30	Feb-86	1	8
Telecomm	Microphone	software	10	Jul-86	2	2
Telecomm	Networking	Telecomm	18	Jun-85	1	1
Unix	CadMac: a high–end Macintosh	analysis	27	Nov-85	1	5
Utility	Disk Caches Compared	software	28	Oct-85	1	4
Utility	Sidekick	software	11	Oct-85	1	4
Utility	Tempo: programming your Macintosh	software	01	Apr-86	1	10
Utility	Acta: outliner tour de force	software	02	Jun-86	2	1
Utility	HFS Backup	software	09	Jun-86	2	1

Project Reports



This section of *Wheels for the Mind* presents many of the projects that are being developed at various colleges and universities. The listings are not restricted to Consortium schools or to schools in this country; you will find projects that are being done at universities all over the world. Project listings that appeared in the previous issue appear here in an abbreviated form; new listings and listings that have been updated are described in more detail (when available), and are indicated by an apple in their heading.

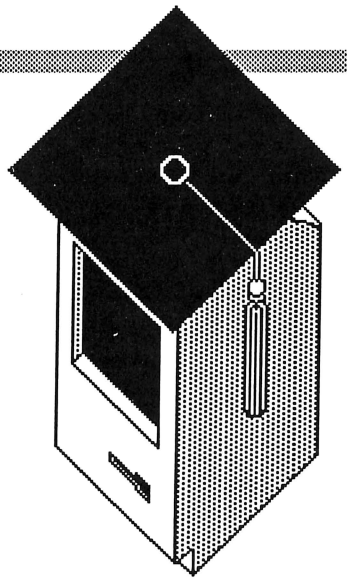
Some of the items listed here are just in the beginning stages, while others are completed and ready for distribution.

Wherever possible, we have tried to list an appropriate contact, in case you are interested in finding out more about a particular listing.

One of the original intents of organizing educational institutions around the Macintosh was to provide opportunities for creative people to share their expertise and talents. We are sure that you will find much of that creativity in evidence here.

Please feel free to collaborate on developing an application, to develop one on your own, or to simply use one that was developed by one of your colleagues.

Most important, let us know what YOU are working on. Perhaps there are others around the world who might like to work with you! Send us a description of your project.



Astronomy

Instrument Panel

Robert Loewenstein of Yerkes Observatory is developing a remote-control "instrument panel," written on a Macintosh 512K in a beta release of Kriya Systems' Neon, for use with a new 3.5-meter telescope to be built in New Mexico. For further information, see the winter 1986 issue of *Wheels for the Mind.*

Current status: Running and under continued development.

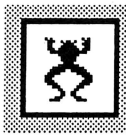
Contact: Robert Loewenstein, Sr. Research Associate, Dept. of Astronomy and Astrophysics, Yerkes Observatory, P.O. Box 0258, Williams Bay, Wisconsin 53191; (414) 236-5468

Spaceflight Instrument Data Collection

The Laboratory for Astrophysics and Space Research is using a Lisa computer to perform bench tests, accelerator calibration, and eventually, data collection from spaceflight instruments that measure cosmic rays.

Current status: In use.

Contact: William Sterner, Lead Staff Analyst, Information Technologies and New Services Computation Center, University of Chicago, 5737 University Ave., Chicago, IL 60637; (312) 962-7172 STAFF.BILL@UCHICAGO



Biology

Botanical Classification

A plant identification program that enables students to identify any one of 90 different shrubs and trees in the local environment. Identifies all flora native to the immediate area around Drexel University.

Current status: Classroom testing.

Contact: Coordinator, Microcomputer Program, Office of Computing Services, Drexel University, Philadelphia, PA 19104

Identifying Biological Specimens

AUTOKEY is an interactive system (written in Absoft's FORTRAN) that is designed to provide quick and accurate identifications of biological specimens for people with little direct experience in biological identification.

Current status: Complete.

Contact: Dr. Ron Hellenthal, Dept. of Biology, University of Notre Dame, Notre Dame, IN 46556

Biology

Use in Biology

The biology department is using Macintosh computers in three areas: document production, teaching, and research. The document production process is fairly standard, using products such as Microsoft Word, Thunderscan and MacVision to write proposals and papers. Faculty and graduate students use these systems.

In teaching, the department currently uses Macintosh computers in teaching introductory genetics and developmental biology. The genetics simulation models allelic variations in populations using the Hardy-Weinberg Law. These simulations are running on top of Microsoft Excel. The developmental biology lesson is a visual simulation of human development, starting from a single cell. This 14-stage lesson is actually a series of movies controlled by the VideoWorks Trainer, from MacroMind. The movie-lessons themselves were developed at Washington University by John Constantino and Dr. Roy Peterson.

In research, the Macintosh is used as a DNA analysis tool using the DNA Inspector, from Textco. DNA sequence data is extracted from locally VAX-resident data bases (such as GenBank) using data-base server programs written in-house. Running on the Macintosh and conforming to the Macintosh user interface guidelines, these servers communicate over AppleTalk to the VAX, request information about a particular sequence, and return that information at some later time to the Macintosh.

Tying these uses together is the AlisaTalk network, from Alisa Systems. This software product, running on top of the Kinetics FastPath AppleTalk-Ethernet bridge, provides a connection between the Macintosh computers on AppleTalk and our VAX/VMS network on Ethernet. The Alisa software provides a true file server, allowing the Macintosh computers to increase their storage space and share files. Alisa also provides a print spooler, used to spool printing from the Macintosh systems (or the VAX) through a VAX to the LaserWriter Plus. We have used its library routines to implement our data base servers. These servers allow us to use the Macintosh as a tool for manipulating information, and the VAX as back-end compute/data base servers. We are continuing work on extending AppleTalk over our installed telephone wiring, and on writing additional servers that use VAX as a back end for the Macintosh.

Current status: Ongoing
Contact: Bruce McDiffet, Department of Biology, Washington University, Campus Box 1137, St. Louis, MO 63130; (314) 889-6860

Biology

Evaluation of Natural Habitats

ERAPT is an interactive information retrieval and analysis system that permits the evaluation of natural habitats and organism assemblages using stored information on the environmental requirements and pollution tolerances of biological species.
Current status: A prototype version of this system is currently running on a Macintosh 512K. A full implementation of the system on the Macintosh XL is planned.
Contact: Dr. Ron Hellenthal, Dept. of Biology, University of Notre Dame, Notre Dame, IN 46556

Labtutor

Labtutor (Lisa Pascal) is a package that uses its own command interpreter to process procedure files into interactive laboratory tutorials that present complicated analytical procedures as simple step-by-step instruction sets.
Current status: Unknown.
Contact: Dr. Ron Hellenthal, Dept. of Biology, University of Notre Dame, Notre Dame, IN 46556

DNA Analysis

The Cornell DNA analysis package, which allows the user to perform various types of pattern-matching analyses on strings of DNA and RNA, has been adapted to the Macintosh.
Current status: Complete.
Contact: Craig McGowan, Computer Center, University of Rochester, 727 Elmwood Avenue, Rochester, NY 14620

Heat Exchange in Animals

This program consists of interactive instructional material on the physical laws governing heat transfer and a simulation of the thermoregulatory strategies adopted by animals.
Current status: Near completion.
Contact: Mike Barlow or Karen Lee, IRIS, Cypress Hall, Stanford University, Stanford, CA 94305

Evolutionary Biology

MacClade draws phylogenetic trees input from data files and allows on-screen manipulation of the trees (moving branches, rerooting, and other functions). The evolution of the organisms' features is traced graphically on the current tree, and can be manipulated, thus providing a flexible tool to analyze character evolution. Various assumptions about character evolution are allowed. Tree lengths are calculated. Local branch swapping is the only automatic tree-searching algorithm. The ability to manipulate trees and to analyze character evolution in a graphic and interactive environment makes MacClade a program useful for both research and the teaching of evolutionary biology.
Current status: First release out June 1986. Second release planned for January 1987.
Contact: Wayne Maddison (graduate student), Museum of Comparative Zoology, Harvard University, Cambridge, MA 02138; (617) 495-2472

Biology

Evolutionary Properties of Genetic Regulatory Networks

The thesis of Kauffman's work is that generic properties, being overwhelmingly probable under random construction and mutation, are products of general structural constraints, rather than of selection. In his simulations, fitness is calculated simply as a function of the number of correct connections relative to a "perfect" network. We provide programs for simulating the evolution of genetic networks under selection and mutation for which fitness is calculated as a function of specific and detailed connectivity properties of the "perfect" network. This more detailed information allows empirical testing of W. C. Wimsatt's "developmental lock" models of developmental constraints as products of "generative entrenchment." (Generative entrenchment is a measure of how many features or traits of the phenotype depend upon a given feature for their proper expression. Simulations have already shown that there are generic thresholds [with respect to gene/connection ratios] at which well-entrenched subnetworks can be maintained by weak selection even against very high mutation rates.)

All programs are written in ExperLisp, which produces natural and compact code. (Kauffman's C code is 1500 lines. The core of ours is 150. With front end and graphics, it is 400.) A simulation of the evolution of 100 organisms for 2000 generations, each with 20 genes and 20 connections, with selection and a mutation rate of .0001 per connection end, runs on a 512K Macintosh in less than six hours. This is roughly 30 times as fast as Kauffman's comparable simulations ran in C on a DEC 11-03. It is likely that ExperLisp code will provide similar advantages for simulations involving populations with genetic and/or spatial structure, and we plan to try these in the near future.
Current status: Ongoing
Contact: This work is being done by Jeffrey Schank, (mail: c/o Committee on Conceptual Foundations of Science, 1126 East 59th Street, University of Chicago, Chicago, IL, 60637; (312) 288-7021) with Prof. William C. Wimsatt, Department of Philosophy, 1050 East 59th Street, University of Chicago, Chicago, IL, 60637; (312) 962-8598

Chemistry

Lesson Writer/Chemistry Lessons

This program contains two chemistry lessons for student use: one on inorganic nomenclature and the other on common formulas. These are created using the Lesson Writer authoring tool. Lesson Writer is also included along with a MacPaint file of laboratory equipment, which can be used for the creation of new lessons, tests, or lab reports.
Current status: Available from ACE (Kinko's Academic Courseware Exchange, 800-235-6919; in California, 800-292-6640).
Contact: Coordinator, Microcomputer Program, Office of Computing Services, Drexel University, Philadelphia, PA 19104

Chemistry

Orbital Mixing

Orbital Mixing depicts the molecular orbitals derived from the 1p orbitals overlapping the PI manner.
Current status: Finished and being used in intermediate-level chemistry courses.
Contact: The Personal Computing Center, 1 McNutt Hall, Dartmouth College, Hanover, NH 03755; (603) 646-3249

Mutiple-Step Diffusion in Solids

Professor Robert Huggin's program demonstrates the effect of a two-step diffusion process on the composition profiles within solids. The simulation can be completed under different environmental conditions, producing different profiles.
Current status: In progress.
Contact: Professor Robert R. Huggins, Department of Material Science, Stanford University, Stanford, CA 94305

MacStereo

MacStereo illustrates the Cahn-Ingold-Prelog stereochemical designation system; presents the correct designation procedure for molecules with one chiral center, using static text and four interactive examples; and presents nine interactive problems of increasing complexity.
Current status: In development.
Contact: Coordinator, Microcomputer Program, Office of Computing Services, Drexel University, Philadelphia, PA 19104

PeriChart

A data base of atoms and elements that uses the periodic chart as a control structure, allowing the user to obtain data for such properties as isotopic masses and abundances, electronegativities, oxidation states, and valence electron configurations.
Current status: Available from ACE (Kinko's Academic Courseware Exchange, 800-235-6919; in California, 800-292-6640).
Contact: Coordinator, Microcomputer Program, Office of Computing Services, Drexel University, Philadelphia, PA 19104

Quantum Chemistry

(1) Particle in one-dimensional box with finite wall.
(2) Simple Huckel molecular orbital.
(3) Ab initio self-consistent field molecular orbital plus configuration interaction.
These are all being cleaned up for easy access by the novice user.
Current status: Used (1) in freshman honors chemistry, (2) with junior physical chemistry students, and (3) with seniors and graduate students. (Absoft's FORTRAN.)
Contact: Dr. Maurice E. Schwartz, Department of Chemistry, University of Notre Dame, Notre Dame, IN; (219) 239-7744

Chemistry

Multiplan Spreadsheets

EquilCalc provides a tool for solving equilibrium concentrations in the general reaction aA+bB=rR+sS, given initial concentrations and the equilibrium constant. AcidBaseTitr generates a plot (within Multiplan) of pH vs. added base for the titration of a weak acid with a strong base. Molmass is a simple molecular weight and percent composition calculator.
Current status: Available from Kinko's Academic Courseware Exchange (ACE).
Contact: Kinko's Academic Courseware Exchange, 800-235-6919; in California, 800-292-6640

Reaction Rates

A simulation in chemical kinetics that computes time-dependent concentrations of a chemical reaction after initial concentrations of reactants and temperature are specified.
Current status: Classroom testing.
Contact: Coordinator, Microcomputer Program, Office of Computing Services, Drexel University, Philadelphia, PA 19104

Vistas in Chemistry (MacVIC)

The goal is to develop highly visual, animated graphics that illustrate chemical and physical phenomena; suitable for first-year courses.
Current status: Unknown.
Contact: Dr. Russ Larsen, Department of Chemistry, Texas Tech University, Lubbock, TX 79409; (806) 742-3057

Organic Molecular Structures

This project explores the feasibility of using the Macintosh to generate complex organic molecular structure to aid in teaching organic chemistry.
Current status: Completed.
Contact: Roy Roberts, Chemistry, College of Natural Sciences, University of Texas, Austin, TX 78712

BenchTop Application: NMR Spectroscopy

NMRSystem allows a Macintosh to run, display, and store NMR spectra from a Varian EM-360 NMR Spectrometer. It was developed by Adam Whiting, a chemistry senior at Reed College.
Current status: Completed.
Contact: Marianne Colgrove, Education and Documentation Coordinator, Development Lab, Reed College, Portland, OR 97202; (503) 771-1112 ext. 277

Chemistry

Molecular Editor

A software-based construction kit for building molecules, crystals, or other structures that can be represented by interconnected points or spheres in three dimensions. You can build or edit structures of over 100 atoms from any element in the periodic table. Three-dimensional versions of Cut, Copy, and Paste allow you to build any molecule from an initial collection of basic structures and functional groups. You can also draw structures as all atoms and all bonds, atoms only, or bonds only; rescale the molecule, or independently rescale atomic radii while leaving bond lengths unchanged, allowing continuous variation from a touching-hard-sphere model to a bonds-only model; rotate a molecule through any angle about any of three orthogonal axes, either by single steps or continuously; and rotate or move a portion of a molecule with respect to another portion. Group theory operations such as inversion and reflection through planes are available. You can also measure the distance between any two atoms, the angle between any two bonds attached to the same atom, and the torsional angle about a bond; change the perspective, or viewing angle, from which the molecule is seen; store and retrieve structures on disk; and open up to 20 files at one time and flip the display manually between them. On a Macintosh 512K you can run through a sequence of open files rapidly (five frames/sec) using an animation feature. You can print the molecule in the active window (on an ImageWriter or LaserWriter printer) directly from Molecular Editor. There is an extensive Help Menu. Sample files of organics, functional groups, inorganics, crystals, simple geometric shapes, and a three-dimensional animation of a chemical reaction are included.
Current status: Version 1.0 completed. Available from Kinko's Academic Courseware Exchange (ACE), 800-235-6919; in California, 800-292-6640.
Contact: Coordinator, Microcomputer Program, Office of Computing Services, Drexel University, Philadelphia, PA 19104

Chemistry Network

The chemistry department at Wayne State University in Detroit has, over the last 12 months, installed an extensive AppleTalk network linking 24 Apple Macintosh Plus computers and 5 Apple LaserWriter Plus printers. The network was wired using 5 Apple custom wiring kits and is over 1,500 ft. in length.
Current status: Completed.
Contact: Dr. Louis Romano; (313) 577-2584

Computer Science

Image-Processing Package

A package has been developed that can handle 64- x 64-dot images of 32 gray levels. Facilities include the manipulation of the gray-level histogram (thresholding, windowing, stretching, equalizing, color reversing) and the application of various masks (edge detection, edge and line enhancement, smoothing, isolate removal, and more). The interface is a mouse-driven, hierarchical set of menus.

The display allows the 32 gray levels to be compressed to 10. A bit map generated to simulate them, or any adjacent 10 levels, can be shown correctly. A relief plot is also available, as are the projections onto the x- and y-axes.

This package is designed primarily as a teaching tool for early problems in computer vision, remote sensing work, and image processing in general. The input files are text files (32 levels denoted by a character in the range 0...9, a...v). It is not difficult to generate such files from other formats, but in particular, we have developed programs that will do so from Thunderscan files.
Current status: Version 1 is working. Further facilities and extensions are in progress.
Contact: Tony Adams, Department of Information Science, University of Tasmania, Box 252C, Hobart, Tasmania, Australia 7001

The Extended Show and Tell Language

The Show and Tell Language (STL) is a visual language for keyboardless programming designed for schoolchildren and novice computer users, and has been described in previous issues of *Wheels for the Mind* and other publications. The extended STL will incorporate list processing capabilities (making STL a picture LISP language), and will incorporate constraint-based problem-solving.
Current Status: Completed and ongoing
Contact: (For STL) Mr. K. Kikawa, Technical Manager, Computer Services Corporation, Shinjuku-ku, Tokyo 160-9, Japan; (03) 344-1811
(For extended STL) Dr. Dan Kimura, Computer Science Department, Washington University, Campus Box 1045, St. Louis, MO 63130; (314) 889-6122

Processing Systems

The Department of Computer Science is teaching a course, "Processing systems," using the Macintosh. The course uses the Macintosh Development System to teach students MC68000 assembly-language programming and its application in a complex computer system. Students learn fundamental algorithms and data structures for numerical computation, address translation, storage allocation, and the control of peripheral devices.

Weekly programming exercises are conducted in a recently established Macintosh laboratory. The labs include basic graphics on the screen, linked list manipulation, interrupt-driven cursor display, and disk manipulation. In addition, each student must propose and complete a final project that incorporates the use of a Macintosh peripheral such as the Microbot MiniMover-5 robotic arm, or New Image Technology's Magic digital camera
Current status: Ongoing
Contact: Dr. Maynard Engebretson, Department of Computer Science, Washington University, Campus Box 1045, St. Louis, MO 63130: (314) 889-6160

Computer Science

Computer Science Graphics

An ongoing project that involves sending Macintosh-generated graphics to a VAX Tektronix 4115 B System. Images are digitized on a Macintosh 512K using the Thunderscan optical digitizer. SCAN files as produced by the Thunderscan software are composed of three parts: a header, a packed binary gray map, and a MacPaint-compatible bit map. The gray map provides 32 gray levels per pixel. These gray-map text files are then sent by modem from the Macintosh to a VAX 11/780 minicomputer. In this environment, it is possible to reconstruct the images on Tektronix terminals with favorable results.
Macintosh—Mark Kotary, Computer Science, and Sam Drogo, Life Science,
VAX Tektronix—Robert Dell, Computer Science
Current status: Ongoing.
Contact: Sam Drogo, Life Science, Mohawk Valley Community College, Utica, NY 13501; (315) 792-5409

Rascal

Rascal is a new Pascal-based language-and-development system designed for the Macintosh, incorporating Editor, Linker, Compiler, and Execution Environment in a single application.
Current status: Completed. Commercially available for $129 (discounts for educational users), from Metaresearch, Inc., 1100 SE Woodward, Portland, OR 97202; (503) 232-1712
Contact: Scott Gillespie or Richard Crandall, Reed College, Portland, OR 97202; (503) 771-1112 ext. 277

Assembler/Editor/Loader

This application was designed to allow efficient programming of Intel 8051/8031/8751 microcontrollers. Using the Macintosh window environment, this package provides instant visual access to all steps of program development. It is ideal for programming serially networked MCS-51-based devices or for high-speed data collection and appliance control via external lab computers.
Current status: Completed. Price: $150.
Contact: Richard Crandall, Reed College, Portland, OR 97202; (503) 771-1112 ext. 277

Pascal Pointers

A tutorial-type package for teaching the use of dynamic variables and pointers in Pascal.
Current status: Available from Kinko's Academic Courseware Exchange (ACE), 800-235-6919; in California, 800-292-6640.
Contact: Coordinator, Microcomputer Program, Office of Computing Services, Drexel University, Philadelphia, PA 19104

LR1 Gen

A Parser/Generator. This program accepts an input grammar and generates Parse tables from the grammar. Options: LR, LALR, Strong Compatible LR, Weak Compatible. LR-Parse tables in human-readable and machine-readable forms.
Current status: In development.
Contact: Coordinator, Microcomputer Program, Office of Computing Services, Drexel University, Philadelphia, PA 19104

Computer Science

UNIX-Based C-Development System

SUMacC is the Stanford/UNIX Macintosh C-development environment, a collection of programs that allows you to cross-develop Macintosh applications on your UNIX system and then download them to the Macintosh for execution. Requires VAX, Berkeley UNIX, or Eunice. A tape containing the SUMacC files is available for $65 plus shipping.
Current status: Version 2 completed.
Contact: Maria Code, Data Processing Services, 1371 Sydney Drive, Sunnyvale, CA 94087; (408) 735-8006

Using the Macintosh in a First Computer Course

The Macintosh has been successfully used as the primary learning tool in a first computer course for about 400 students.
Current status: Completed and continuing.
Contact: James Gips, Computer Sciences, Boston College, Chestnut Hill, MA 02167

MacGNOME

MacGNOME is a Macintosh environment for teaching beginning programming.
Current status: Beta test.
Contact: Rob Chandhok, Carnegie Mellon University, (412) 268-3452

Project: Turing Machine

Philosophy Professors Jon Barwise and John Etchemendy have developed programs that are used to illustrate aspects of "computability." One program consists of a flowchart construction kit that enables the students to solve a particular problem by drawing a flowchart. The completed flowchart then drives a simulated Turing machine that executes the program. A second program, Tarski's World, allows the students to test the truth or falsity of statements about a block world that is displayed on the screen. The computer evaluates these statements, which are written by the student in predicate calculus, with respect to the state of the world.
Current status: Available from Kinko's Academic Courseware Exchange (ACE)
Contact: ACE, 800-235-6919, or in California, 800-292-6640

MacProlog

An implementation of Prolog for the Macintosh ported from a PDP11 UNIX version. Version 1 is a straight port with a single, unsizable window. Version 2 will have multiple windows and control menus. Version 1 is available to Consortium members free of charge. Send us a disk.
Current status: V1 is completed. V2 status unknown.
Contacts: Casper A. Boon and Damian P. Meyer, Department of Computer Science, University of Western Australia, Stirling Highway, Nedlands 6009, Western Australia, Australia; (09) 380-2282

Design

3D Geometry

Mechanical engineering professor Douglass Wilde's program allows manipulation of computer-generated pictures as part of instruction in three-dimensional computer-aided design. This CAD program helps design and engineering students visualize solid objects. The basic or orthogonal and isometric projections are built-in. Additional features include scaling, rotation, translation, and hidden-line removal.
Current status: Completed.
Contact: Mike Barlow or Karen Lee, IRIS, Cypress Hall, Stanford University, Stanford, CA 94305

PADL 1—Computer-Aided Design

PADL 1 is a solid-modeling program currently running on VAX computers.
Current status: Unknown.
Contact: Rich Marisa, University of Rochester, 727 Elmwood Avenue, Rochester, NY 14620

Perspective Drawing Recognition

The School of Design and Environmental Analysis is developing a teaching program for recognizing correct perspective drawing.
Current status: Unknown.
Contact: Kate MacGregor, Cornell University, Ithaca, NY

Energy Diagramming

The University of Oregon's Department of Architecture has developed a program that enables students to design buildings that take advantage of the sun's heat and other thermodynamic principles. The software (which has three primary components) allows the designer to draw an intuitive design using a MacPaint/MacDraw-like environment while being free of the technical calculations necessary to determine the energy efficiency of the building. Using a data base of dimensions and calculations, the program provides room measurements based on factors such as climate, the amount of glass, and the location of the building in relation to sun exposure. It also gives the designer calculated feedback on whether or not the rooms and structures are energy efficient. The second part of the program contains lab exercises that the students can complete to learn principles such as wind and sun movement and heat transference. The third part of the program is a library of architectual examples that allow the user to examine, for instance, a building by Frank Loyd Wright. An additional feature of this library is its ability to select a part of the building and enlarge it for close-up analysis.
Current status: Completed.
Contact: G.Z. Brown, University of Oregon, Eugene, Department of Architecture, Eugene, OR 97403; (503) 686-5647

Economics

Problem Sets for Introductory Economics

Covers topics ranging from supply and demand to international trade. Students receive problem sheets for which the instructor assigns random values (within set parameters) for all variables. The instructor's disk generates variable-values answers for problems using these variables. Requires Multiplan.
Current status: Classroom testing.
Contact: Coordinator, Microcomputer Program, Office of Computing Services, Drexel University, Philadelphia, PA 19104

Problems in Macro- and Microeconomics

A five-disk set that includes microeconomics—introduction, cost functions, demand, competitive models, and wages (pure competition, imperfect competition, monopsony, and union), macroeconomics—money and banking, international exchange and trade, economic objectives, income determination, and fiscal policy, and an economic data base. Requires Multiplan and MacWrite.
Current status: Classroom testing.
Contact: Coordinator, Microcomputer Program, Office of Computing Services, Drexel University, Philadelphia, PA 19104

Education

MacSED—Software-Evaluation Data Base for Macintosh

Current status: Unknown.
Contact: Prof. Michael Scriven, Education, University of Western Australia

Mentor

This project will use Odesta Helix, and possibly other software, to provide a question-answering, CAL, and testing data base for teachers.
Current status: To be completed in November 1986.
Contact: Prof. Michael Scriven, Education, University of Western Australia, Stirling Highway, Nedlands 6009, Western Australia

An Examination-Answering Program

This application is for use by first-year computer science students; it is part of an automatic examination-generation, answering, and marking system.
Current status: Completed.
Contact: Ranadeva Peries, Department of Computer Science, University of Western Australia, Stirling Highway, Nedlands 6009, Western Australia

Education

Future Vision

The Advanced Technology Learning Center (ATLC) at Texas Tech University was developed to help students learn and to help faculty improve their courses by using microcomputer technology. The 25,000-square-foot facility allows hands-on learning, experimentation, teaching, research, and curriculum and research development.

Housed in this facility is the Classroom 2000, a futuristic educational setting using Macintosh computers and LaserWriter printers. University professors can reserve the entire room for use by their classes for presentations. Other uses of this unique lab environment include training courses on specific software, and open access use for any university student or teacher.
Current status: Completed and ongoing.
Contact: Dr. Lee Alley, Assistant Vice President for Computing and Communications, Texas Tech University, Box 4120, Lubbock, TX 79409; (806) 742-1527

Residence Halls

The University of Illinois has set up four clusters of 8 to 12 Macintosh computers in residence halls on the Champaign/Urbana campus. Twelve students have been trained by Claudia Wintergerst, the Apple marketing support representative, to use the Macintosh computers and several applications such as MacWrite and MacDraw. These first group of 12 students act as peer trainers to train other students in the residence halls to increase their productivity through microcomputing.
Current status: Ongoing.
Contact: Howard Diamond, (217) 333-1300, or Donna Price, Project Coordinator, (217) 333-0770

Residence Halls

Northwestern University has set up the first Macintosh Lab in a residence hall on the campus of the residential College of Commerce and Industry, for business school students. With 15 Macintosh computers, 1 Macintosh XL, and several ImageWriter printers, the lab is heavily used by approximately 40 students per night. It was set up to provide more computer access for students and to provide another tool for their coursework. As a result of the success of the first lab, Northwestern is looking ahead at setting up other labs in residence halls across campus.
Current status: Ongoing.
Contact: Gene Shook, Student Coordinator, (312) 492-1998, or Don Collins, Provost and Project Manager, (312) 491-5255

Education

Residence Halls Computing

The University of Michigan's Computing Center's Information Technology Division and Housing Office have been involved in a team effort known as the Residence Halls Computing Project, or ResComp for short. It was started in September 1985 as a pilot project to computerize one residence hall and in January of 1986 was expanded to all residence halls.
Current status: Ongoing.
Contact: Ellen Hoffman, Computing Center (313) 936-2712, and Marvin Parnes, the Housing Office, (313)764-7408.

College Administration

The Office of the Dean of Students in the College of the University of Chicago has proposed the computerization of its student advising and management information activities through the implementation of a network of Macintosh Plus workstations. The advisors need to quickly grasp and process vast amounts of information from diverse sources, including their own files and the Student Information System (SIS). Currently, 15 full-time academic advisors and other more specialized resource people are responsible for policy administration and advising students on courses, programs, and academic opportunities. In addition, they handle course registration, as well as resource coordination among Admissions, Financial Aid, the Registrar, Housing, Career Placement, Alumni, and Academic Divisions. Most of these activities are presently being handled through paper systems. The strategic goals for the project are to recover time for the college advisors and other support personnel; enhance the search for academic honors, awards, and aid; and improve communication and scheduling.

The general approach is incremental. We expect to implement the system function by function, with continuous testing of hardware, software, programs, and procedures by potential users. With this approach, we hope to reduce the risk of large, costly mistakes while constructing a solid foundation for full office automation. An essential aspect of the project is that it arises directly from the users themselves. By developing their own systems the advising staff will also be determining their own goals, needs, equipment, software, and training. The functions we have identified to date are as follows:

- Filing and sorting information
- Scheduling—for both students and administrators
- Communication—among advisors and with other offices
- Word Processing—for both internal and external distributions
- Budgets
- Analysis—of student information, and of workloads and effectiveness
- Registration—the major administrative function of the office

For each function, small pilot projects will develop prototype data bases and procedures with appropriate software (probably Helix) and hardware. These pilot projects will take place over the next two quarters as a first phase in the project. The results of the first phase will then be developed into a plan with a work schedule and budget for a second, full implementation phase. Macintosh Plus computers will form the core for each workstation, along with a hard drive

College Administration

(possibly SCSI), and local ImageWriter II printers. An initial goal will be to develop access to a Microdata Sequel (PICK OS) data base through terminal emulation. This may evolve into Amdahl and DEC interfaces, possibly based on an Ethernet backbone or the University's IBX digital phone system. AppleTalk will be used for clustered printing and eventually for shared data-base access to a minicomputer or possibly to an Apple-based file server.

As mentioned before, a major aspect of the implementation plans is that they are being developed largely by the system's prospective users, rather than by outside computer experts. The advisors are ardent generalists and proud computing novices. They are also experts in their own work. The extreme friendliness of the Macintosh interface, coupled with the business power represented by the Macintosh Plus, makes it especially appropriate for this project. The microcomputing innovations pioneered by the Macintosh have generated the radical idea that users can themselves build a system that fits their work, rather than accept a system for which they must adapt their work. This project will provide an interesting test of that idea.

This does not mean, however, that we don't want help from others. If anyone has already worked toward computerizing dean of students' activities, please let us know of your work so that we can gain from your experience and share ours with you.

Current status: Completed.

Contact: B. Patrick Hall, Dean of Students' Office, University of Chicago, Harper 233, 1116 E. 59th Street, Chicago, IL 60637; (312) 962-8622

Engineering

Earthquake Engineering

Reviews and defines basic principles of earthquake seismology through text and graphic displays. A short review of selected applications of seismology is included.
Current status: Classroom testing.
Contact: Coordinator, Microcomputer Program, Office of Computing Services, Drexel University, Philadelphia, PA 19104

TK!Solver for Problems in Materials Science and Engineering

Includes rapid crack propagation in plastic pipes, prediction of phase diagrams, determination of purity of materials, polymer flow and processing, and impact behavior of materials. All important behavior-of-materials properties (thermophysical, mechanical, electrical, chemical, and so on) are covered.
Current status: Classroom testing.
Contact: Coordinator, Microcomputer Program, Office of Computing Services, Drexel University, Philadelphia, PA 19104

SISO

An application for the study of linear dynamic systems and linear control theory. Includes simulations, designs, and graphic output. Systems are designed and analyzed in both time and frequency domains. Eigenvalues, eigenvectors, frequency response, and root locus.
Current status: Classroom testing.
Contact: Coordinator, Microcomputer Program, Office of Computing Services, Drexel University, Philadelphia, PA 19104

Transforms, Windows, Modulation

Follows the same user-interaction format and screen-design philosophy as the Signal Operations module for undergraduate courses in systems engineering. Performs discrete Fourier Transform of signals using the Fast Fourier Transform (FFT) algorithm and allows a variety of signal operations in both the time and frequency domains.
Current status: Available from Kinko's Courseware Exchange (ACE), 800-235-6919; in California, 800-292-6640.
Contact: Coordinator, Microcomputer Program, Office of Computing Services, Drexel University, Philadelphia, PA 19104

Geotechnical Engineering

This package of nine separate programs handles traditional geotechnical engineering problems that are very calculation oriented. These programs are self-directed and titled as follows:

1. Settlement due to surcharge fill.
2. Settlement due to footing loads.
3. Interpretation of consolidation data.
4. Interpretation of triaxial data.
5. Slope stability by total stress analysis.

Engineering

Geotechnical Engineering

6. Slope stability by effective stress analysis.
7. Slope stability by using geosynthetics.
8. Bearing capacity of shadow foundations.
9. Bearing capacity of deep foundations.

Current status: Classroom testing.
Contact: Coordinator, Microcomputer Program, Office of Computing Services, Drexel University, Philadelphia, PA 19104

Structural Analysis

This software package contains 12 interactive programs for the analysis of determinate structures. Topics covered include equilibrium, determinacy, shear and moments for beaming and frames, truss analysis, and deformation of structural members. This software is designed as supplemental material for the first course in structural analysis.
Current status: Classroom testing.
Contact: Coordinator, Microcomputer Program, Office of Computing Services, Drexel University, Philadelphia, PA 19104

Simulation Design

Professor Jucker of the Department of Industrial Engineering has designed graphical exercises for analyzing random-number generators.
Current status: Completed.
Contact: Mike Barlow or Karen Lee, IRIS, Cypress Hall, Stanford University, Stanford, CA 94305

Signal Operations

For use in introductory systems engineering courses, this module offers a variety of signal-generation options ranging from singularity functions, sinusoids, and exponentials to arbitrary functions. Versions 1.0 and 1.3.
Current status: Available from Kinko's Courseware Exchange (ACE), 800-235-6919; in California, 800-292-6640.
Contact: Coordinator, Microcomputer Program, Office of Computing Services, Drexel University, Philadelphia, PA 19104

Deformation and Fracture of Materials

Professor Miller of Materials Science is developing a program to model deformation and fracturing of materials under a variety of conditions.
Current status: Completed.
Contact: Mike Barlow or Karen Lee, IRIS, Cypress Hall, Stanford University, Stanford, CA 94305

Engineering

Systems Analysis

An interactive, menu-driven software package that uses Microsoft BASIC. Can be used as an aid to solving problems involving matrix manipulations.
Current status: Completed.
Contact: Doug Starfield, (317) 494-6256

Introduction to the Thermosciences

The Rankine and Brayton cycles are tutorials written for Engineering 32, "Introduction to the Thermosciences," by Professor Robert Eustis. They are used as the prime instructional medium for the topics, replacing lectures. Each lesson consists of two types of screen presentations: instructional frames and interactive frames. In the first type, content material is presented. In the second, the student attempts to solve a problem by generating graphs and data for different thermodynamic systems.
Current status: Completed.
Contact: Mike Barlow or Karen Lee, IRIS, Cypress Hall, Stanford University, Stanford, CA 94305

Bode Analysis Program

This program provides a magnitude-and-phase plot for a linear system with a rational transfer function.
Current status: Completed.
Contact: T. J. Wagner, Department of Electrical and Computer Engineering, University of Texas, Austin, TX 78712

Transmission-Line Simulation

This program shows voltages and currents on a transmission line with real source and load impedances. This type of input, the length of the line, the attenuation, and the value of the source and load impedance may all be selected.
Current status: Completed/Ongoing.
Contact: C.H. Roth, Department of Electrical and Computer Engineering, University of Texas, Austin, TX 78712

Wave Guide Lab

The Mac488 interface was used to interface a multimeter to the Macintosh. Microsoft BASIC was used to control the experiment and graphically display the data acquired.
Current status: Complete.
Contact: John Lefor, Electrical Engineering, University of Rochester, 727 Elmwood Avenue, Rochester, NY 14620

Engineering

Aerodynamics

This project is an attempt to demonstrate that most computer programs in the computational fluid dynamics area can be successfully implemented on a Macintosh with two 400K disks, 512K RAM, and hard-copy computer graphics capability.
Current status: Completed.
Contact: George Dulikravich, Penn State University

Desktop Laboratory

This program incorporates Macintosh systems for initial use in the "Computer Applications in Chemical Engineering" course, and will form the nucleus of a considerably larger number of applications in other chemical engineering courses.
Current status: Completed.
Contact: Eugene Wissler, Chemical Engineering, College of Engineering, University of Texas, Austin, TX 78712

Graphics Course

A revised curriculum for a junior-level computer graphics course in the electrical and computer engineering department will be designed to incorporate the unique graphics capabilities of the Macintosh.
Current status: Unknown.
Contact: Terry Wagner, Electrical Engineering, College of Engineering, University of Texas, Austin, TX 78712

Macintosh Use in Engineering Instruction

The University of Iowa College of Engineering has standardized on Macintosh computers. Every faculty member has a Macintosh on his or her desk. One student lab houses 40 Macintosh computers and another identical lab is currently being set up. In total, there are more than 160 Macintosh computers in use, along with ImageWriter and LaserWriter printers. As part of a three-year curriculum and computer integration plan, the computers are being used in teaching programming and engineering courses, and for project work in every aspect of the engineering curriculum. More than 100 Macintosh computers are available for student use, either as stand-alone units or configured using MacApollo software (developed by the University of Michigan) as an interface to the Apollo minicomputer for computer-aided design (CAD) work.

In a major software development effort, faculty members are responsible for software development in their curriculum areas. Additional commercial software such as Macintosh Pascal and public-domain software developed at other universities has been adopted. Soon, each student will be expected to use the computer for a minimum of 10 hours per week.
Current status: Ongoing.
Contact: Dr. Jon Kuhl, Faculty Director, Iowa Computer Aided Engineering Network (ICAEN), University of Iowa, 2402 Engineering Building, Iowa City, IA 52242; (319) 353-7214

Engineering

Logic Application

TRUTHTAB is a Macintosh program for printing truth tables from logical expressions. It can handle classical (two-valued) logic, as well as multivalued logic with a small number of truth values. It is written in standard UCSD Pascal and does not use any features unique to Macintosh.

LOGICLAB is the next version of TRUTHTAB. It will allow logics with an infinite number of truth values. It will even employ inexact or fuzzy reasoning in the form of a "jump to conclusions" feature. Since there is no general agreement on semantics for multiple-valued logic, LOGICLAB will allow the user to define operator semantics using mathematical expressions.
Current status: TRUTHTAB is finished and listings are available.
Contact: Ed Nilges, 1000 8th Ave., A203, Seattle, WA 98104

Steamprop

Eliminates time spent by students in the interpolation of steam tables. By using this program, problems in the optimization of thermal systems that were previously too time consuming or required the use of mainframe computers may now be solved. Steamprop fixes thermodynamic states in both US and SI units and has a range from 32°F (0°C) to 1500°F (800°C) and up to 5000 psia (340 bars). Over this range it agrees with the 1984 NBS/NRS steam tables to better than 1 part in 200; in the subcritical range the agreement is better than 1 part in 200, making this program suitable not only for instrumental use but also for commercial applications.
Current status: In testing
Contact: Coordinator, Microcomputer Program, Office of Computing Services, Drexel University, Philadelphia, PA 19104

Chemical Engineering

A program to model heat transfer in a human finger has been written in Microsoft BASIC (2.0 or 2.1) for a graduate course in Biomedical Engineering. The program considers the decrease in blood flow toward the end of the finger and also accounts for countercurrent transfer between arteries and veins. At the end of the finger, the series-parallel model of Mitchell and Myers (*Biophys. J.* 8:897, 1968) is used.
Current status: Preliminary version finished, but further development ongoing.
Contact: Ken Solen, Chemical Engineering, 350-G CB, Brigham Young University, Provo, UT 84604; (801) 378-6237

Chemical Engineering

A program in Microsoft BASIC (2.0 or 2.1) to model the clearance of barbiturate from the body using an artificial kidney. The program, written for a graduate course in biomedical engineering, considers the body in a four-compartment model with blood and tissue subcompartments in equilibrium as described by Dedrick and Bischoff (*Chem. Eng. Prog. Symp. Ser.* 64:32, 1968). Transfer equations are solved using a fourth-order Runga-Kutta algorithm.
Current status: Preliminary version finished, but further development ongoing.
Contact: Ken Solen, Chemical Engineering, 350-G CB, Brigham Young University, Provo, UT 84604; (801) 378-6237

Engineering

Chemical Engineering

A simple function-plotting program has been written in Microsoft BASIC (2.0 or 2.1) to plot any function provided in a subroutine. The program makes use of windows, buttons, and pull-down menus to allow easy selection of the plot features. The resulting plot can be reproduced on the ImageWriter with full-page resolution or can be transferred to other graphics programs (such as MacDraw) for further enhancement.
Current status: Essentially complete.
Contact: Ken Solen, Chemical Engineering, 350-G CB, Brigham Young University, Provo, UT 84604; (801) 378-6237

Chemical Engineering

A curve-fitting program is being developed in Microsoft BASIC (2.0 or 2.1) to fit a set of x-y data to linear, quadratic, cubic, or exponential functions to a curve and plot the data and/or the fit. The program makes use of windows, buttons, and pull-down menus to allow easy selection of the plot features. The resulting plot can be reproduced on the ImageWriter with full-page resolution or can be transferred to other graphics programs (such as MacDraw) for further enhancement.
Current status: Essentially complete.
Contact: Ken Solen, Chemical Engineering, 350-G CB, Brigham Young University, Provo, UT 84604; (801) 378-6237

Industrial Engineering

Professor Henry Riggs has designed Multiplan templates for use by students in an industrial accounting class. The students complete exercises on the principles of industrial accounting and then apply what they have learned to complete a major financial accounting exercises at the end of the course.
Current status: Completed.
Contact: Professor H. Riggs, Department of Industrial Engineering, Stanford University, Stanford, CA 94305

Mechanical Engineering

During the last three semesters, mechanical engineering students have developed a user/tutorial package for the Macintosh (works best on 512K). The software, called MacHEATRAN, is very user-friendly and is designed for use in conjunction with any common heat-transfer textbook. The programs are menu driven and include the following 15 sections:

1. Introduction to conduction, convection, and radiation as covered in the first chapter of many texts.
2. Steady state conduction through composite walls. All three coordinate systems.
3. Fins—called "extended surfaces" in most texts.
4. Conduction shape factors for 24 shapes.
5. Convection heat transfer coefficient and friction in duct flow.
6. Convection heat transfer coefficients for flow across single tubes and tube bundles, including pressure drop across tube bundles.
7. Free convection, external, and enclosed flows.

Mechanical Engineering

8. Radiation configuration factor.
9. Flat plate heat transfer coefficient and boundary layer thickness for both free and forced convection and both laminar and turbulent flow.
10. Heat exchangers, using a combination of LMTD and the effectiveness-NTU method.
11. The electrical resistance network method (Oppenheim's) for solving radiation problems for net heat transfer between two, three, or four surfaces where the geometry and surface properties are known.
12. Lump capacity system analysis where up to six lumps represent the problem. Any of the lumps can be connected to a constant temperature source or a constant temperature sink, and a heat source can drive any or all of the lumps.
13. Uniform heat generation for steady state in any of the three common coordinates.
14. A lumped capacity program set up to solve just the same type of problems solved by the Heisler charts; again, all three coordinate systems.
15. A linking portion of a program is set up so the user can add a favorite application program to the MacHEATRAN disk, called TRY YOUR SKILL.

Some fluid properties are included in the software and can be called automatically by the name of the fluid. The program is in Microsoft BASIC 2.0.

This software is designed for the beginning heat-transfer student or as a refresher for the rusty student. It is a tutorial disk, but is also appropriate for people who do an occasional heat-transfer estimate in their engineering work. Experts in heat transfer will probably find this program too simple. With user's manual, the package is available for $95 and site licenses for $295.
Current status: Completed.
Contact: Richard Ulrich, Mechanical Engineering, 242-J CB, Brigham Young University, Provo, UT 84604; (801) 378-6542

Mac Controls

Mac Controls is an interactive educational and modeling program for engineers who are working with the design and control of systems. Mac Controls allows input of equations for a single closed-loop system in a domain, without any limitation as to the degree of polynomials in each transfer function. Menu choices permit plotting of phase or gain, using either logarithmic or linear interpolation for the range of desired frequency. Unlike comparable mainframe programs, Mac Controls is fast—5 seconds to plot! By *Wheels* press time, the program should also include an option to plot root locus for a system; addition of three-mode controller (PID); and output phase and gain margin. Mac Controls is written in FORTRAN.
Current status: Completed and ongoing.
Contact: Menachem Rafaelof, University of Utah Computer Center 3116 MEB SLC, UT 84112; (801) 581-6802

Engineering

MACVLSI

This program is for use in learning application-specific integrated circuits, and allows the student to do trial circuit layouts on the Macintosh. Completed designs are then downloaded to the VAX host computer for design-rule checking and circuit extraction. Student imagination can be put to work in the quick evaluation of alternatives in circuit layout.
Current status: Classroom testing.
Contact: Coordinator, Microcomputer Program, Office of Computing Sciences, Drexel University, Philadelphia, PA 19104

Geology

Project: Plate Tectonics

This program, developed by Professor Michael McWilliams, performs the extensive calculations involved in plate tectonics and displays the results graphically. The students can view a plate "history," possibly using a "What if?" mode, or they can construct their own scenarios with novel geological and geophysical data as input.
Durrent status: In progress.
Contact: Professor M. McWilliams, Department of Geophysics, Stanford University, Stanford, CA 94305

Geology Simulation

The main idea of this simulation program is to move a fluid element through a basin and to calculate the processes of transport, deposition, and erosion in dependence on the fluid velocity at a certain section of the basin. In addition, currents and the compaction of sediments can be simulated. There are a number of input parameters that control the system: initial fluid velocity, fluid volume, sediment input, grain-size distribution, physical properties of the sediment, and the initial basin configuration. The program is set up in modules and supports an extension to three dimensions. It offers a variety of possible experiments: simulation of growth faults, horst and graben structures, oscillating input parameters, randomly generated fluctuations of any input parameter, and more. The simulation also includes a number of sophisticated display programs. The simulated section can be displayed with its facies distribution, time lines, grain-size distribution, and layer display. Another graphics routine allows display of enlarged sections of particular parts of the simulated model. The program has been tested and it showed very realistic results, suggesting that this model can be used for sedimentation simulations.
Current status: Completed.
Contact: Klaus Bitzer, Geologisches Landesamt, Albertstr.25, 78 Freiburg, West Germany

Engineering

Groundwater Hydrology

Reviews and defines basic principles of groundwater hydrology and geology (including steady-rate well hydraulics).
Current status: In development.
Contact: Coordinator, Microcomputer Program, Office of Computing Services, Drexel University, Philadelphia, PA 19104

Structural Geology

Uses basic properties of igneous rocks (mineralogy, color, texture, and so on), metamorphoric rocks (mineralogy, texture) and sedimentary rocks (mineralogy, grain size, fossils) to assist in the identification of common rocks. Properties determined in the laboratory service as input. Reviews basic principles of structural geology and applies those principles to the interpretation of geological maps.
Current status: Classroom testing.
Contact: Coordinator, Microcomputer Program, Office of Computing Services, Drexel University, Philadelphia, PA 19104

History and Politics

Treaty of Versailles

A series of exercises to teach the major territorial and reparation decisions in the Treaty of Versailles. Uses questioning and interactive map exercises.
Current status: Available from Kinko's Academic Courseware Exchange (ACE); 800-235-6919, or in California 800-292-6640.
Contact: Coordinator, Microcomputer Program, Office of Computing Services, Drexel University, Philadelphia, PA 19104.

Historical Simulation: The Would-be Gentleman

A sophisticated simulation of economic and social life in 17th-century France, which casts the student as a nobleman. The aim is to maximize wealth and prestige over the two generations of the simulation.
Current status: Available from Kinko's Academic Courseware Exchange (ACE).
Contact: ACE, 800-235-6919; in California, 800-292-6640

Humanities

Tools for Writers

A text-analysis program that flags all uses of "to be" and occurrences of anomalous or incorrect words and phrases within a customized dictionary, among other things.
Current status: Available from Kinko's Academic Courseware Exchange (ACE).
Contact: ACE, 800-235-6919; in California, 800-292-6640

BRS Simulator

Software that simulates the search environment of BRS, a commercial data-base vendor. It is used to introduce freshmen to data-base concepts and search strategies before going on-line.
Current status: Available from Kinko's Academic Courseware Exchange (ACE); 800-235-6919; in California, 800-292-6640
Contact: Coordinator, Microcomputer Program, Office of Computing Services, Drexel University, Philadelphia, PA 19104

Macintosh Writing Center

Whittier College has established a Macintosh-equipped writing center. Its purpose is to provide the college's 1,050 undergraduates with access to Macintosh microcomputers, a variety of word-processing software, and printing facilities for their college writing. The center contains 12 networked Macintosh computers running MacServe, a LoDown 20 hard disk, two ImageWriter II printers, and a LaserWriter printer. The Writing Center is open to all students and faculty.
Current status: Ongoing.
Contact: Professor Charles Adams, English Department, Whittier College, Whittier, CA 90608; (213) 693-0771

Writing Lab

This project explores the feasibility of using the Macintosh in a writing laboratory environment.
Current status: Completed.
Contact: Jerome Bump, College of Liberal Arts, University of Texas at Austin, Austin, TX 78712

Computer Programming as a Liberal Art

Goal Statement: The goal of this course is to combine the techniques of computer programming with liberal arts modes of thinking about the human meaning and social applications of computers.

Course Description: The course will approach computer programming both as a specialized set of techniques tied to a particular subject matter, and as a set of interdisciplinary arts adaptable to a wide range of problems. Thus, students will learn how to write programs in Pascal on Apple Macintosh computers, while at the same time learning how to reflect upon the activity of programming in more

Computer Programming as a Liberal Art

open-ended terms. Course assignments will include programming problems, several short quizzes, a midterm exam, and a paper. The course will be taught by William Sterner of the computation center and Don Crabb of computer science.

In specific, the course will present traditional techniques of problem solving, algorithm choice, program coding, and debugging through lectures and programming assignments—as does any introductory course. In addition, the course will examine these techniques as interdisciplinary arts of "recovery," "invention," "presentation," and "systematization."

Such arts acquired in the context of programming can then be generalized and adapted to other subjects. This additional part of the course will be approached through assigned readings, class discussion, and a "reflective" paper on the human meaning and social implications of computer programming.

Because of this wider context, the students must also learn to recognize those situations where even liberal programming arts are not adaptable to some problems. If such situations were assimilated to a "programming" approach, they might present potential dangers for society and might waste technological resources.

Prerequisites: Since the challenge of this course will lie in doing a lot of clear thinking in a humanistic mode, the student should have at least a rough sense of humanistic reflection. Previous coursework in the common core or philosophy would prove valuable. No prior programming experience will be assumed, and no mathematics beyond high school will be required.

Intended Students: This course is intended especially for liberal arts students who want to explore the meaning of computer technology, and for beginning computer science students who want to understand the broader significances of their discipline from the outset. Students with extensive programming or mathematical background should not register for this course.
Current status: New course offered summer 1986.
Contact: William Sterner, Lead Staff Analyst, University of Chicago Computation Center, 1155 E. 60th Street, Chicago, IL 60637; (312) 962-7172; STAFF.BILL@UCHICAGO, A113

Automated Language Processing

Selected faculty at Michigan State University are using Macintosh personal computers with MacProof from Automated Language Processing Systems (ALPS) to help teach English composition.

There are five designated campus sites including the English department, American thought and language department (the freshman writing program), the Learning Resource Center, and the library. Students composing papers in one of MSU's Macintosh/ALPS labs have their writing automatically diagnosed

Automated Language Processing

problems with grammar and syntax, puncuation, spelling, even racist and sexist language. More than 1,000 students are expected to benefit from MacProof each semester.

The project, designed to enhance the creative writing process, demonstrates the impact that computers have on the creative processes, specifically on writing skills. Results of this impact will be surveyed, and the project will be expanded to include additional students and other departments on campus.
Current status: Completed.
Contact: Dr. Peter Lyman, Assistant Director of Academic Computing, 315B Computer Center, Michigan State University, East Lansing, MI 48824; (517) 353-0722

Languages

Macedonian-English Dictionary

Current status: Unknown.
Contact: Prof. R. De Bray, Department of Slavonic Languages, Australian National University

Hypertext/Hypermedia

The Institute for Research in Information and Scholarship has undertaken a long-term research project to design and develop a hypertext-hypermedia system. The research is being conducted within the context of Brown's Scholar's Workstation Project—a decade-long effort directed at creating a campuswide network of high-powered, graphics-based workstations for use in research and teaching in the sciences and liberal arts.
Current status: A long-term, ongoing project. An initial hypertext/hypermedia prototype allows users to create links to interactive videodisc frames as well as text.
Contact: Norm Meyrowitz, IRIS Scholar's Workstation Group Manager, Brown University, Providence, RI

Drill, an English-Language Drill Program

Drill can be used in almost any course where text- and/or picture-based drilling is useful. The disk includes Drill, (the program), Create a Drill (a script about writing drills), documentation, and sample drills.
Current status: Available from Kinko's Academic Courseware Exchange (ACE).
Contact: ACE, 800-235-6919; in California, 800-292-6640

Languages

Language Teaching Using a "Bitworld"

Students learning Russian use a program developed by Richard Steele that associates vocabulary items with graphic representations of objects. If an object is selected, the corresponding vocabulary item is displayed (either alone, as part of a paradigm, or in a sentence). Some objects, such as cupboards, can be "opened" to reveal their contents, thereby giving access to a new set of objects.
Current status: In progress.
Contact: Mike Barlow, IRIS, Cypress Hall, Stanford University, Stanford, CA 94305

The Glossarizer

The Glossarizer consists of a simple editor for entering text and a "glossary mode" for attaching notes, translations, or other explanatory information to a word or phrase in the original. It was designed as a tool for foreign-language teaching.
Current status: Completed.
Contact: Katy McKinin, Rice University, ICSA, Box 1892, Houston, Texas; (713) 527-4986

Spanish Language

The purpose of this program is to develop software programs designed to help bilingual-education students improve their written language skills in Spanish; to provide students with hardware and Spanish-language software in a laboratory setting; to gather data on the effectiveness of this type of computer-assisted instruction; to revise the software accordingly; and to use the software for in-service training of current bilingual-education teachers.
Current status: Unknown.
Contact: George Blanco, Curriculum and Instruction, College of Education, University of Texas at Austin, Austin, TX 78712

Demotic Dictionary

Dr. Jan Johnson of the University of Chicago Oriental Institute is using the Macintosh to produce a dictionary of the Demotic language. The dictionary staff has already compiled about 6 megabytes of material to be stored on a hard disk for easy access and analysis. Dr. Johnson plans to print the dictionary as a supplement to Erichsen's Demotisches Glossar on the LaserWriter, if the required fonts can be created for the device. If not, the dictionary will be printed on an ImageWriter and then offset.
Current status: Ongoing.
Contact: Jan Johnson, Director, Oriental Institute, 1155 E. 58th Street, Chicago, IL 60637

MacLang

MacLang is an authoring system that allows language teachers inexperienced in computer use to write computer-assisted language learning (CALL) exercises on a Macintosh 128K microcomputer. With the MacLang authoring system, teachers can create software specifically tailored to their curriculum and to the needs of their students, rather than being obliged to teach new grammar and vocabulary so that students can use prepackaged commercial software. In its

Languages

MacLang

current version, MacLang offers the possibility of preparing five types of exercises using English and any one of the following languages: French, Spanish, Italian, Portuguese, German, Rumanian, or Russian. The five types of exercises are as follows:

1. Vocabulary-equivalency drills in which students must give the target-language equivalent of the English word presented on the screen.
2. Vocabulary-definition drills in which students must type the target-language word that fits the definition presented on the screen in either English or the target language.
3. Fill-in-the-blank exercises in either English or the target language, based on both grammar points and vocabulary.
4. Question-and-answer exercises, in which the student user types a complete answer in response to a question asked by the computer.
5. Multiple-choice questions of any type.

Current status: Ongoing.
Contact: Judith Frommer, Department of Romance Languages and Literatures, Boylston 227, Harvard University, Cambridge, MA 02138

Word Study in Greek and Latin

The University of Minnesota Microcomputer Systems Group, in cooperation with the classical studies department, has developed two computer-assisted instruction programs for studying commonly used words and their Greek and Latin roots. The Basic Word Study in Greek and Latin program presents a series of drills and exercises dealing with common prefixes, suffixes, and roots in English derived from Greek and Latin. Basic Word Study in Greek and Latin helps freshmen and sophomores increase their vocabulary and improve their writing skills; it might be used by majors in classical civilization, Greek, and Latin, and may be of interest to students of Greek and Roman literature, archaeology, and culture.

Another program developed by the same group, Advanced Word Study in Greek and Latin, presents a series of drills and exercises dealing with an advanced development of vocabulary and verbal skills, and etymological and word-building principles; it also includes a historical background of English. This program might be used by students of the School of Nursing, the College of Pharmacy, and programs in physical and occupational therapy; it might also be used by students of Greek and Roman literature, archaeology, and culture, and by majors in classical civilization, Greek, and Latin.

Both programs are independent study tools that run on the Macintosh computer. The programs are user friendly: they display instructions and questions in Macintosh dialog boxes, comment on the student's answers, and reprompt for incorrect responses. The student's interaction with the computer is simple and straightforward, and only minimal knowledge of computers is needed. While running the programs, the student can stop a particular drill, choose another drill, or exit.

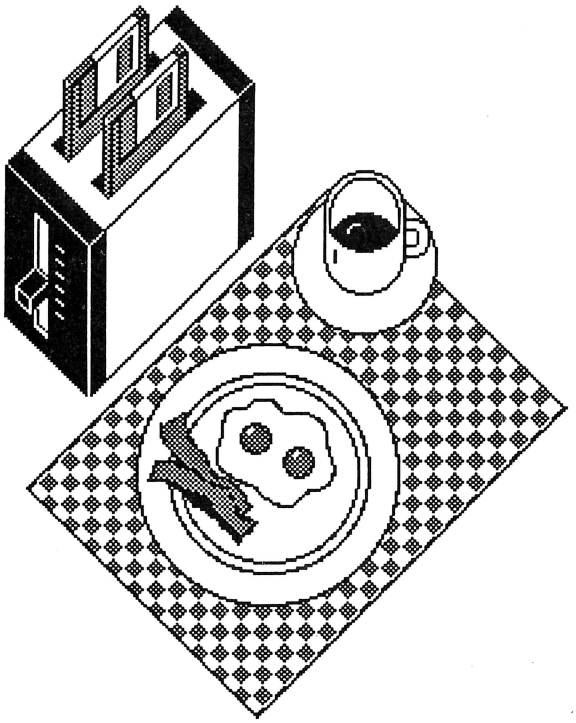
Languages

Word Study in Greek and Latin

Current status: Both programs are completed. They run on the Macintosh 512K and Macintosh Plus, with one or two disk drives. A user manual is supplied with each program. An optional study guide that provides preparatory material for the computer drills and exercises is also available. The study guide may be reproduced. Retail price of each program with the study guide and user manual: $69.95.
Contact: University of Minnesota, Department of Classical Studies, 310 Folwell Hall, 9 Pleasant St. S.E., Minneapolis, MN 55455; (612) 625-5353

Russian Verbal Aspect Drill

A program to drill students in selecting the correct verbal aspect, person, tense, and mood to insert in Russian sentences. Currently there are 200 sentences (10 for each of 20 verbs). Students may call for a translation of the Russian target sentence or the conjugation of the verb being drilled at any time. Incorrect answers may be retried as many or as few times as the student desires. A help screen containing a synopsis of the basic rules governing aspectual usage is always available.
Current status: Compiled Pascal version available.
Contact: Martin P. Rice, Professor of Russian, Coordinator of Humanities Computing Development, University of Tennessee, Knoxville, TN 37996; AppleLink: U0317



Medicine

Neurobiology: Waveforms and Filters

This program produces a high/low pass filter that is fully controlled by the Macintosh mouse.
Current status: Completed and available on Creative Solutions MacFORTH Demo Disk.
Contact: Bruce Land, Cornell University

Analysis of Photomicrographs of Brain Cell Aggregates

Stephen Sellers has written a Microsoft BASIC program for the image analysis of photomicrographs of brain cell aggregates.
Current status: Running and available by request.
Contact: Stephen Sellers, Department of Pharmacology and Physiology, Abbott Hall 318, 947 E. 58th Street, Chicago, IL 60637; (312) 962-9331

Ventilator Simulation

The program, designed by Professor Kohatsu, helps medical students learn about ventilator management. It computes and displays three kinds of information: data from the ventilator/lung processes, and recommendations or predictions.
Current status: Complete.
Contact: Mike Barlow or Karen Lee, IRIS, Cypress Hall, Stanford University, Stanford, CA 94305

Electrolyte Disturbances and Treatment

Professor Maffly's program combines both instruction and testing of topics in the biochemistry of water and electrolytes.
Current status: Completed.
Contact: Mike Barlow or Karen Lee, IRIS, Cypress Hall, Stanford University, Stanford, CA 94305

Review of Human Physiological Defense Mechanisms

Faculty at the University of Utah College of Nursing are developing a computer-assisted tutorial for review of content pertaining to the immune, inflammatory and coagulation systems.
Current status: In development.
Contact: Beryl M. Peters or Sandra H. Haak, College of Nursing, University of Utah, 25 South Medical Drive, Salt Lake City, UT 84112

Medicine

Greek and Latin Terminology in the Medical and Biological Sciences

The University of Minnesota Microcomputer Systems Group, in cooperation with the Classical Studies Department, has developed two computer-assisted instruction programs for studying medical and biological terms in Greek and Latin. The Greek and Latin Terminology in the Medical and Biological Sciences program presents a series of drills and exercises dealing with roots and terms in Greek and Latin basic to the medical and biological sciences. The Greek and Latin Terminology for Nursing program presents a similar series, particularly oriented to the vocabulary of nursing. Both programs might be used by students of the School of Nursing, programs in physical and occupational therapy, and the College of Pharmacy; they might also be used by students of Greek and Roman literature, archaeology, and culture, and by majors in classical civilization, Greek, and Latin. While running the programs, the student can stop a particular drill, choose another drill, or exit.

Current status: Both programs are completed. They run on the Macintosh 512K and Macintosh Plus, with one or two disk drives. A user manual is supplied with each program. An optional study guide that provides preparatory material for the computer drills and exercises is also available. The study guide may be reproduced. Retail price of each program with the study guide and user manual: $69.95.

Contact: University of Minnesota, Department of Classical Studies, 310 Folwell Hall, 9 Pleasant St. S.E., Minneapolis, MN 55455; (612) 625-5353

Animated Anatomy Tutorial

Students at Washington University Medical School are using a self-study course in human anatomy to depict cell development. The program shows a single cell developing in the womb from a single fertilized egg up to 12 weeks. This development depicts the evolution of the cell into external and internal surfaces of the body, and how these surfaces are then transformed into systems. A dissection on screen demonstrates surfaces and organ systems of the developing embryo. Other topics available on disk are the development of the digestive tube system and the development of the nervous system. Programs covering the circulatory system and genital-urinary system are currently under development. The animation provides three-dimensional instruction at a fraction of the cost of film.

Current status: Completed and ongoing.

Contact: Dr. Roy Peterson, Department of Anatomy, Washington University Medical School, 660 S. Euclid, St. Louis, MO 63110; (314) 362-5000

Scan Analysis

Scan Analysis was developed to provide a low-cost image-processing system for biological and biochemical scientists. The system employs the ThunderScan digitizer, which digitizes autoradiograms, polyacrylamide gels, and more. The scan file obtained from ThunderScan is then processed using the Scan Analysis program. Three different modes of analysis are available: free-format area selection, area averaging and integration, and two-dimensional scanning and integration. The data obtained from Scan Analysis have been shown to

Medicine

Scan Analysis

correspond to the actual radioactivity being measured by autoradiography, and to actual protein concentration by dye binding. Coupled with ThunderScan, Scan Analysis brings quantitative image processing to scientists at a very low capital outlay.
Current status: Unknown.
Contact: Timothy Burcham, Department of Neurobiology, Stanford University School of Medicine, Stanford, CA 94305; (415) 497-5811

Programs for Ecological and Statistical Analysis

Current status: Unknown.
Contact: A. Cockburn, Department of Zoology, Australian National University

Diagnosing Maladies

The College of Veterinary Medicine at Mississippi State University is taking a new approach to diagnosing maladies. Faculty and students have harnessed high technology to enable them to arrive at the most accurate and complete diagnoses possible. Using the Macintosh computer and software program called "Problem Knowledge Coupler," the causes, findings, and diagnostic tests results are correlated. Take the example of feline anemia: There are 58 diseases that can cause anemia in a cat and 316 relevant findings (historical questions, diagnostic results, and so forth) that impact those diseases. Therefore, in this single diagnosis, there are over 18,000 interrelationships that need to be considered to arrive at the appropiate treatment. There is no way the human mind can retain that amount of information, but the computer can. Four coupler programs—including such as Feline Anemia and Canine Cough—are already in use at the school. Additional programs specific to different species are under development.
Current status: Completed.
Contact: Dr. Philip Bushby, Program Director, College of Veterinary Medicine, Mississippi State University, MS 39762-5521; (601) 325-3432

Personal Choice

Personal Choice—A Guide to Selection of Birth Control, is a program developed by the Brown University Health Services center. It's the first in what is hoped will be a series of health education software aimed at a college-age audience. The program includes an illustrated review of reproductive anatomy and physiology, and an interactive section where the user answers questions about health history, lifestyle, and personal priorities. The program makes great use of the Macintosh interface with only one page of user documentation required. It's been very well-received, because it prepares a person for understanding the issues involved in making a decision about birth control.
Current status: Completed.
Contact: Anne M. Andrade, MSNP or Richard Linden, Associate Director at the Brown University Health Services; (401) 863-3953

Medicine

Anesthesiology

The anesthesiology department at Duke University is using Macintosh in the operating room to analyze blood pressure and oxidation level information. They track the data, enter it into an Excel spreadsheet, and then graph the trends.

The Medical School is also adding a Macintosh lab and will be transferring data between Macintosh computers and IBM PCs. They've also developed custom applications on the Macintosh using FORTRAN.
Current status: Completed.
Contact: John Lippert, Apple Higher Education Executive

Ekteach

Teaches interpretation of the electrocardiogram and tests students with respect to the names of the EKG deflections, the amplitudes, and the time relationships of the normal EKG reading
Current status: Completed
Contact: Dr. Stephen Dublin, Department of Biomedical Engineering, Drexel University, Philadelphia, PA 19104

Cancer Research

Dr. Thea Tlsty is a very well-known and respected researcher at the University of North Carolina. Working in the Lineberger Center at the university, she is studying the causes of cancer at the DNA level. She uses the Macintosh for data input in the research lab, as well as Business Filevision for inventory control and explanations of equipment and various chemicals and solutions.
Current status: Completed.
Contact: John Lippert, Apple Higher Education Executive

Miscellaneous

Drexel Plot

A graphing package designed to plot on a grid data pairs entered from the keyboard or via a SYLK file.
Current status: Available through Kinko's Academic Courseware Exchange; 800-235-6919, or in California, 800-292-6640.
Contact: Coordinator, Microcomputer Program, Office of Computing Services, Drexel University, Philadelphia, PA 19104

Miscellaneous

BenchTop Products for the Sciences

The chief software products are The BenchTop, which creates a powerful laboratory network system, and Rascal (see description under "Computer Science"), a real-time Pascal. These applications allow the user to define and operate on-screen simulations of actual laboratory apparatus. The hardware products are a Universal Microcontroller, a laboratory single-board computer based on Intel microprocessors; and the BenchTop Instrument, a customizable, shielded instrument box. Both hardware devices talk to the Macintosh according to a rigorous protocol.
Current status: Universal Microcontroller completed. Consortium price: $500. Retail price: $750.
Contact: Metaresearch, Inc., 110 SE Woodward, Portland, OR 09702; (503) 232-1712

Lesson Writer

An authoring tool that enables the user to create short lessons using basic MacWrite procedures and any graphics that can be transferred via the standard Clipboard.
Current status: Completed
Contact: Available from Kinko's Academic Courseware Exhange (ACE), (800) 235-6919; in California, (800) 292-6640

Mainframe Print Servers

As a value-added service provided to SEAGate, Macintosh users will be able to print their documents remotely using a VAX or Pyramid Computer as a print server for the LaserWriter or other PostScript machines. In addition, they will be able to convert MacWrite documents to troff to allow addition of technical features using Eqn.
Current status: Unknown.
Contact: Craig McGowan, Computer Center, University of Rochester, 727 Elmwood Avenue, Rochester, NY 14620

Electronic Chalkboard

This project uses a Macintosh as a novel electronic chalkboard, permitting drama students in directing classes to illustrate body/eye focus and movement patterns of actors in harmony with recorded voices.
Current status: Completed.
Contact: Bernerd Engel, Drama, College of Fine Arts, University of Texas at Austin, Austin, TX 78712

Drexel Disk

The Drexel Disk uses an interactive map of the Drexel University campus to orient new Macintosh users to the microcomputer-related facilities available. Send a self-addressed, stamped disk.
Current status: Available from Drexel University.
Contact: Coordinator, Microcmputer Program, Office of Computing Services, Drexel University, Philadelphia, PA 19104

Miscellaneous

ColorPaint

Imagine painting a brilliant sunset using 16 million colors! Students at Reed College are learning about color theory as well as creating masterpieces using a Macintosh computer, a Vectrix color graphics machine with a large display monitor and a software product called ColorPaint. The artists can paint with 20 tools—including three-dimensional shading—on a screen that can display 512 colors at any one time. It's a compelling visual display of the power of personal computers and graphics. Photographs can be taken of the screen and then reproduced. With the Macintosh interface, the program is easily used by novices, artists, media students, and scientists. ColorPaint was developed using Rascal, a programming language for the Macintosh computer created at Reed College.
Current status: Completed.
Contact: Dr. Richard Crandall, Technology Plan Director, Reed Collge, Portland, Oregon 97202; (503) 771-1112 ext. 255

Creative Art on the Macintosh

In the School of Creative Arts at San Francisco State University, the blending of art forms is taking place through the use of Macintosh computers. Each one of the eight departments in the School of Creative Arts owns a Macintosh. In addition, there are up to 15 Macintosh computers in a unique interdisciplinary computer lab setting devoted to projects of the very creative students and their professors. Since September, a true meeting of the arts has occurred in the lab, where students of dance, music, film, theater, arts, broadcast communications, and the Center for Experimental Interdisciplinary Art have joined together to work on projects involving choreography, animation, costume design, and more. Besides using current commercial applications such as MacPaint and MacDraw, faculty and students are also developing their own software. The collaborative projects promise infinite possibilities for awakening aesthetic senses.
Current status: Ongoing.
Contact: Dr. August Coppola, Dean, School of Creative Arts, San Francisco State University, 1600 Holloway, San Francisco, CA 94132; (415) 469-1471

Physical Education Lab

The College of Physical Education operates a Macintosh lab with 10 Macintosh computers, 9 external drives and 3 ImageWriter printers. The lab is open to students currently enrolled in physical education courses. It is used primarily by students in PE 191 for their computerized weight training and by graduate students using NCSS to do statistical analysis of performance levels.
Current status: Completed.
Contact: Tom Catherall, Recreation Management and Youth Leadership, 279 RB, Brigham Young University, Provo, UT 84604; (801) 378-4991

KeyMouse — A Mouse-Driven Keyboard

The mouse-driven keyboard (KeyMouse) is a desk accessory program compatible with 128K, 512K, and Macintosh Plus computers. It requires no special

Miscellaneous

KeyMouse — A Mouse-Driven Keyboard

system, and runs parallel with any program that supports desk accessories in the normal manner.

The KeyMouse DA allows handicapped users equipped with head-mounted mouse hardware to perform all keyboard functions, including typing, with an on screen version of the keyboard. The user types a character or makes a selection by moving the cursor to it and clicking. The Caps Lock, Shift, Option, and Command keys of KeyMouse are latching or "sticky" keys, allowing combinations such as Option-G or Option-C to be entered. The program also updates the system keyboard memory so that Option-click and Shift-click functions can also be performed. KeyMouse also works with dialog boxes. The physical keyboard can be used in conjunction with the KeyMouse screen keyboard. Except for repeating keys, almost anything the physical keyboard can do, the Keymouse DA can do.
Current status: Completed.
Contact: John Halleck, University of Utah Computer Center 3116 MEB SLC, UT 84112; (801) 581-6802

Data Base and Classification of Minoan and Mycenaean Seals

A data base of Minoan and Mycenaean seals is being developed. The pilot scheme includes only those seals that show human figures (about 500 of the 5,000 published seals). In conjunction with the data base, a classification scheme for the seal designs is being developed. The data base contains images of the seals, as well as the factual information and classification.
Current status: A data base has been set up. It is being tested, and the classification scheme is being refined. Next year, we hope to input the majority of the data.
Contact: Tony Adams or Janice Crowley, Department of Information Science, University of Tasmania, Box 252C, Hobart, Tasmania, Australia 7001

Template for Experiments in Facial Expression Research

An expert-systems knowledge base, called Expresser, is being developed using MacWrite and MacKit Level 1 from Knowledge Systems Environments. This system is based on Paul Ekman's research with facial expressions. According to Ekman, humans exhibit 43 different facial expressions. Through the use of this knowledge base, one can identify what emotion a person might be exhibiting. The expert system also contains a large collection of digitized graphic images that are incorporated into the knowledge base. This expert system is going to be used in an experiment that will measure human reactions to facial expressions. MacKit Level 1 is required to use the MacWrite Template.
Current status: First version of Expresser to be completed in December 1986. All experimental materials will be available in May 1987.
Contact: Paul J. Giguere, University of Lowell, 1 University Ave., Box 5622, South, Lowell, MA 01850

Miscellaneous

Camera

Camera is an educational program that graphically simulates the process of finding the proper exposure using a camera's light meter, and dynamically demonstrates the relationship between f-stop, shutter speed, and film speed. As the user changes the settings on the simulated camera controls, a digitized picture changes density to represent the effect that that change would have on a slide made at that exposure. The picture may also be viewed as a negative to demonstrate the effect of exposure on negative density.
Current status: Completed.
Contact: Craig Hickman, Assistant Professor of Art, Department of Fine and Applied Arts, University of Oregon, Eugene, OR 97403

Scenic Design Drawings and Simulations

In Phase I, scenic design drawings are created by the designer on the Macintosh using MacDraft. These drawings are manipulated and amended on disk by the technical director and lighting designer to develop light plots and detailed working drawings. Additionally, a catalog of ground plan and section drawings of particular theatres is being created to speed other designers and design students in the creation of drawings for future productions.

Phase II focuses on the creation of two-dimensional theatrical design simulations. This will be accomplished by digitizing researched images or preliminary sketches using video and optical digitizing devices (Thunderscan and additional video digitizer). The initial images will be further altered and combined using MacPaint and ArtGrabber. Additionally, due to the compatibility of images created with these programs, complex scene shifts can be graphically illustrated in real time to other designers, directors, and technicians, using VideoWorks.
Current status: Phase I complete, Phase II ongoing.
Contact: Robert N. Schmidt, Department of Drama, University of Texas at Austin, Austin, TX 78712; (512) 471-5341

Miscellaneous

PCal — a Pocket (and other) Calendar Maker

If you're one of those totally organized folks to whom their "Pocket Planner" is second only to their 27-function digital stopwatch, PCal may just be the program you've been waiting for. PCal lets you simply and easily create your own planner and other styles of calendar for printing on the ImageWriter or LaserWriter printer. The program allows the designer to enter over 500 personal events (birthdays, anniversaries, meetings, and conferences) and local holidays in addition to the 16 preprogrammed holidays. On-line Help gives instructions for entering and modifying dates and events.

With PCal you can create the following calendar types:

TYPE	SIZE	ORIENTATION
Year	8.5 x 11	Landscape or Portrait
Month	8.5 x 11	Landscape or Portrait
Week on a page	4.25 x 5.5	Portrait only
Week on two pages	4.25 x 5.	Landscape only
Day on a page	5.5 x 8.5	Landscape only
Day on a page	8.5 x 11	Portrait only
Day on two pages	5.5 x 8.5	Landscape only
Day on two pages	8.5 x 11	Portrait only

Calendars can be printed on both sides of the paper so they can be assembled into "book" form with the pages in proper order. For pocket calendars, the pages must be cut, stacked, and folded to produce the finished product.
Current status: Completed.
Contact: Joseph F. Buchanan, University of Utah Computer Center 3116 MEB SLC, UT 84112; (801) 581-6802

Music

Mozart

Mozart offers music students with no formal composition experience the opportunity to experiment with musical arrangement, tempo, articulation, and dynamics, allowing them to compose a minuet from any of six predefined two-measure phrases written by Mozart.
Current status: Available from Kinko's Academic Courseware Exchange (ACE).
Contact: ACE, 800-235-6919; in California, 800-292-6640

Music

MIDI Recorder/Editor

Robert Allen, a student at the University of Colorado, is writing an application that records and edits events generated by MIDI (Musical Instrument Digital Interface) instruments. Event files (or compositions) are played back by a Yamaha DX-7 and a Yamaha TX-216 in real time as performances.
Current status: Running and under continued development.
Contact: Robert Allen, 1300 30th Street, Apt. E3-22, Boulder, Colorado 80303; (303) 786-0642

Baroque Guitar Data Base

To date, over 2,000 separate melodies have been entered into the system, with each entry including author, title, source, page, meter, time signature, key, length, melody (in the original key), melody (transposed to the key of C), type of piece (allemande, courante, prelude, and so forth), correspondences in other sources, and additional notes of interest.
Current status: Ongoing.
Contact: Craig H. Russell, Department of Music, California Polytechnic State University, San Luis Obispo, CA 93407

Simulation of 18th-Century Organ

A simulation of an 18th-century organ in order to illustrate the musical articulation and phrasing of the time. The program, devised by Professor Houle of the music department, will produce sound and musical notation.
Current status: Completed.
Contact: Mike Barlow or Karen Lee, IRIS, Cypress Hall, Stanford University, Stanford, CA 94305

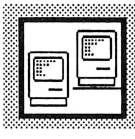
Fundamentals of Music

A program is being developed for instruction in the fundamentals of music. The first module, which should be completed by the end of summer, deals with intervals and harmony. It has two unusual features: it makes use of sampled sound to provide more realistic tones for interval recognition, and it is not limited to the 12 pitch classes of equal temperament. It will be smart enough to drill students interactively on interval spelling and recognition in either the standard equal temperament or one of several others, but it will also be possible to use it for experimentation—to hear the sounds of various intervals and harmonies as they would be interpreted in different systems of intonation. One of the on-screen keyboards has its claves divided into regions so that a particular enharmonic spelling may be selected. The program is called Practica Musica and is primarily intended for university music education.
Current status: Ongoing.
Contact: Jeffrey Evans, Department of Music, University of California, Santa Barbara, CA 93106

Music

Musical Slide Show

With no programming instruction, users can write music on the screen (notation that can be printed out), hear the score played through the Macintosh's speaker (or an external system), add text and graphics as required, and edit the results to form a "slide."
Current status: In development.
Contact: Coordinator, Microcomputer Program, Office of Computing Services, Drexel University, Philadelphia, PA 19104



Networks

AppleTalk Network

Darterminal provides network services to Macintosh computers on an AppleTalk network (internet). It runs protocols specific to Dartmouth's Kiewit network and will not run elsewhere without substantial work.
Current status: Completed (improvements continuing).
Contact: Jim Perry, Kiewit Computation Center, Dartmouth College; (603) 646-2643

SEAGATE

SEAGATE is a network "gateway" between Ethernet and AppleTalk that allows UNIX to provide servers for Macintosh AppleTalk applications.
Current status: Completed.
Contact: Maria Code, Data Processing Services, 1371 Sydney Drive, Sunnyvale, CA 94097; (408) 735-8006

Washington University Campuswide Network

Washington University is building a campuswide network capable of transmitting pictures between workstations throughout the schools of liberal arts, engineering, medicine, and six other professional schools.
Current status: Continuing.
Contact: Jerome Cox, Ph.D., Chairman, Dept. of Computer Science, Washington University, 509 Bryan Hall, St. Louis, MO 63130

Agriculture-related Information System

Macintosh computers link the 67 offices in the Pennsylvania State University Cooperative Extension Service to the university mainframe, providing county agents and extension faculty with the latest information on agriculture-related issues. Called the Pennsylvania Extension Network (PEN), the on-line information and communications system also enables farmers and other residents

Agriculture-related Information System

throughout the state to communicate electronically via a personal computer and modem. The network has helped in emergency situations, such as a recent outbreak of salmonella; within hours a public statement and a seven-page background report on the disease reached every county. Also, in many non-emergency situations information has reached farmers, pathologists, county agents, and Pennsylvania residents in a matter of minutes. Additional services provided by this project include development of software programs for Apple Macintosh computers that focus on agriculture, administration, 4-H youth, and family living. The Milk Market Comparisons, Land Rent Calculator, and Dairy Barn Ventilation Sizing are just a few of the programs already available. Many of the programs come in the form of templates.
Current status: Completed and ongoing.
Contact: Dr. Graham E. Bell, Support Services Manager, Cooperative Extension Service, 344 Agricultural Administration Building, University Park, PA 16802; (814) 863-3449

Instructional Science

The Instructional Science Department has a LaserWriter and ImageWriter on a faculty AppleTalk network, as well as a graduate lab with AppleTalk network and data-switch connections to campus mainframes. The department has been experimenting with a Macintosh programming class using TML Pascal to teach elements of the Macintosh interface—windows, menus, QuickDraw, and so on.
Current status: Completed.
Contact: Paul Merrill, Instructional Science, 201-B MCKB, Brigham Young University, Provo, UT 84604; (801) 378-7072

Physical Education Projects

The College of Physical Education has two Appletalk networks. One, used in the college offices, connects six Macintosh Plus systems (three with LoDown 20 SCSI hard disks), a LaserWriter Plus, and two ImageWriter II printers, and is served by a Bernoulli Box 10+10 network server. The three hard disks will soon be connected to the network with TOPS to make it possible to access any hard disk from any Macintosh. This net is used primarily for printer sharing, hard-disk backup, and some data sharing for software development. The other network connects four Macintosh Plus systems that are used by the college programmers for software development. Currently being developed:

- A program to control inventory for the college. The College of Physical Education has hundreds of thousands of pieces of equipment. To better control and inventory the equipment, this program will use a bar-code reader to first read equipment users' student information from their student ID card, then the equipment ID (basketball, racquet, football, net) from printed bar codes in the equipment checkout room. The program will ask for the number of the specific item (basketball #23, racquet #15). In this way, lost or damaged items can be immediately traced. When an item becomes broken, a precise equipment usage report can be generated by the program to see how many hours (or days

Physical Education Projects

or minutes) that particular item has been used, and this can be compared with the product warranty. Expected completion: fall 1986.

- A program to run the 1988 NCAA Track and Field Finals, to be held at Brigham Young University. A smaller version of this program (using Omnis 3) is complete and was already used successfully at BYU when it hosted the Utah 3A High School Track and Field Championships. The high school program schedules each contestant's heat and lane. Upon finishing, the times (or distances) are entered, then the program automatically calculates the winner and rankings, schedules the atheletes in the next heat and assigns them a lane based on an algorithm that puts the fastest runners in the fastest lanes. This information is available immediately for coaches, atheletes, press, announcers, and so on. At the NCAA Finals, 20 or so Macintosh computers will be networked using a Corvus OmniNet: one system in the announcer's booth, one at the finish line, one at the long-jump pit, one at the pole-vault pit, and so forth. Data entry clerks will sit at each Macintosh and enter scores interactively as they happen. All other users will have immediate access to the scores by simply pulling down a menu that describes the event; the current standings will be listed. Expected completion: winter 1987.

- All PE 191 (weight training) students use a computerized training system that prescribes the appropriate exercises, weights, and repetitions for each student based on a test that determines the one-repetition maximum of each of six basic exercises. The software plans out training courses in general fitness, body building, and power training, and includes three phases in each course. Every three or four weeks, the program changes the training plan to include more weight, fewer repetitions, and a change in exercise. Training programs can be individually altered according to changes in student circumstances. Each student enrolled in PE 191 must buy the $12 program, which uses the runtime Microsoft BASIC interpreter, and Macintosh computers are made available in the PE Macintosh lab.

Current status: Unknown.
Contact: Bruce Holley, Chairman, Physical Education Services, 112 RB, Brigham Young University, Provo, UT 84604; (801) 378-3994
Barry Jenkins, 1972 N. Main, Orem, UT 84606; (801) 224-6834

IAS Lab/Workshops

Brigham Young University Instructional Applications Services conducts weekly free workshops on Macintosh software for faculty staff and students. The workshops are taught in a specially built lab with ten Macintosh computers. Workshop instructors have found that workshop attendees are often so fascinated by the Macintosh interface (Control Panel, puzzle, clicking and dragging, and so on) that they neglect to listen to the instruction and go home no more proficient than when they came, so a method was worked out to solve this problem. One Macintosh is the master, and the other nine are slaves. The instructor has a panel of switches that control the video output of each slave. In this way, the instructor can "take over" the screen display of every slave (or selected individual slaves) so that students see exactly what is on the master's screen, and don't "doodle." This also makes it easy for the instructor to demonstrate a certain

concept, and then give control back to the students to practice it on their own. The master has a video output port for projection onto a lab screen. The lab is also available for individual rental (for as low as $0.50/hour bulk) and faculty can reserve it for use by their classes.
Current status: Completed.
Contact: Harold Hendricks, Computer Teaching Services, 2330 HBLL, Brigham Young University, Provo, UT 84604; (801) 378-2647

IAS Lab/Workshops

Computer Applications Services serves the entire campus with the Computer Consultation Center (CCC). Open 8 to 5 weekdays, the CCC provides a walk-in, hands-on environment where students and faculty have access to most major computer periodicals, a large collection of public domain software, and more. Programs can be tried out in the center itself or checked out overnight. The availability of software for checkout (with documentation) has been a convenient incentive to refrain from making illegal copies of software when "test-driving" it for suitability.

Computer Consultation Center

Consultants make office calls on campus to solve site-dependent problems. In a separate location, two consultants man a telephone bank to answer call-in questions on any computer subject from mainframe to Macintosh. In front of each "phone consultant" is an IBM PC and a Macintosh with a LoDown 20 SCSI hard disk. This gives the consultants virtually instant access to the same software the caller is using, so they can work through the problem together.

The CCC staff hold weekly training meetings so that each consultant can become proficient on most major software packages. CCC computers and printers are available for rent from 5 P.M. to 8 P.M.—$1/hr for computers, 10¢/page for printing.

The CCC also runs a free bulletin board service with FIDO software on an IBM PC AT with 4Mb RAM, and 40Mb hard disk. Both IBM PC and Apple Macintosh interest areas are supported with public domain software and Macintosh news and information.
Current status: Completed.
Contact: John Edwards, Technical Coordinator, Computer Consultation Center, 169 TMCB, Brigham Young University, Provo, UT 84604; (801) 378-7699

YNET

Brigham Young University's central campus network (called YNET) connects twelve mainframe computers: six VAX 750s, two VAX 8600s, three IBM 4341s, and an NAS 9000. Many campus departments use the Macintosh as (among other things) a terminal to computers on YNET, with the help of specialized software and hardware. Access to the IBM and NAS mainframes is through a data switch that connects directly to a HYDRA 3270 protocol

Networks

YNET

converter, which converts ASCII character streams, allowing the Macintosh to emulate a locally attached 3270.

VaxToPC and PCToVax are VAX file transfer programs that do data transfer between the VAX 8600 and personal computers. These programs allow for MacTerminal 1.1, XModem, ASCII, and VTerm file transfer protocols.

The KERMIT protocol is supported on all campus mainframes.

"MacShare" is an electronic-mail service provided on YNET for Macintosh enthusiasts. Any Macintosh-related information, rumors, or questions are mailed electronically to an address called "MacShare." Then network software distributes the message to all people on the MacShare distribution list. A message tells users that there is mail waiting for them the next time they log on to YNET (or immediately, if they are already logged on). YNET is a part of BITNET, so BYU Macintosh users have almost instant access to other Macintosh users all over the world.
Current status: Completed.
Contact: **YNET**—Jim Eggett, Information Systems Services, 167 TMCB, Brigham Young University, BrighamYoung University, Provo, UT 84604; (801) 378-5025
PC to VAX/VAX to PC—Jim Logan, Information Systems Services, 403 CB, Brigham Young University, Provo, UT 84604; (801) 378-6551
MacShare—John Taylor, Harold B. Lee Library, 3008 HBLL, Provo, UT 84604; (801) 378-6278

Evaluation of Macintosh and IBM PC Local Area Networks

At Argonne National Laboratory, which is operated by the University of Chicago for the U.S. Department of Energy, computing services has completed a project to evaluate, demonstrate, and recommend standards for personal computer local area networks. We analyzed the use of personal computers at Argonne and developed a plan to limit our investigation of local area networks to those networks that provide connections for Apple Macintosh and IBM personal computers. We identified functional requirements that we considered necessary to integrate personal computers into the laboratory's hierarchy of computing resources and set the scope of the project at a level that would allow completion of the evaluation in a reasonable time frame (several months).

Computing services evaluated several commercially available alternatives based on the technical requirements and decided on the phased implementation of two local area networks. In Phase I, two networks were installed. We implemented the AppleTalk Personal Network from Apple Computer to connect eight Apple Macintosh systems, an Apple LaserWriter, and a TriData Communications Server (a gateway for IBM 3270-type terminal emulation access to IBM host computers). We also installed a 3Com EtherSeries Ethernet-based network to connect seven IBM PCs, a 3Com network server, a Hewlett Packard LaserJet printer, and a Bridge Communications Server. During Phase II, we installed a

Evaluation of Macintosh and IBM PC Local Area Networks

gateway between the two networks in order to provide similar capabilities to Macintosh and IBM PC users on the networks and to enable the exchange of PC-DOS and Macintosh files. We installed the 3Com EtherMac products as a gateway between the networks and to extend the EtherSeries shared disk and printer capabilities to the AppleTalk network. The Phase I installation began in October 1985 and we completed the evaluation in February 1986. A summary of the project was published in an Argonne document (Technical Memorandum 438) in March 1986.

Computing services' experience during the project led to the following conclusions:

- Local area networks can increase the usefulness of personal computers by allowing resources to be added incrementally at small marginal costs.
- Local area networks involve costly initial investments for cabling and shared resources.
- Local area networks can increase the speed of some personal computer operations by making high-performance devices available.
- Manpower and expertise are required to design, install, and manage these networks as well as to install applications software and train and assist users.
- User acceptance of the networks was good overall. The more sophisticated personal computer users found the networks to be useful tools that expanded their capabilities. Less experienced users needed more guidance and training to appreciate the new capabilities.
- Most local area networks lack the sophisticated communications to transfer files to host computers outside the local network.
- The availability of multiuser software for personal computer networks may be a limiting factor in establishing a successful network.

Argonne's networking goal is a not-too-local, local area network. Argonne is a campus-type environment. As local area networks are installed within buildings, the need arises to communicate among different networks within the campus. A laboratorywide network is thus required. Such communications will involve networks of various protocols: DECnet, TCP/IP, XNS, AppleTalk, NJE, and SNA. Ideally, users will be able to send and receive files transparently, as if they were all physically connected to a single network. Argonne's computing services plans to continue working toward this goal.
Current status: Phase I completed, additional work ongoing.
Contact: Linda Winkler, Computing Services, Argonne National Laboratory, 9700 South Cass Ave., Argonne, IL 60439

Networking

The Harkness Project was set up as a pilot project to network computing services at Brown University. The students join a cooperative organization that allows them to manage the computing services themselves. Development at Brown has resulted in the creation of a wiring hub that extends AppleTalk and allows Macintosh and IBM PC computers to connect to network devices. Each

Networking

dorm room has a twisted pair and coaxial cable connection that gives all students access to laser printing, hard-disk storage, and high-speed mainframes. The Harkness Library is now a computer center where students can go to work and study. It's the preferred working and learning center and has promoted a true learning collaboration among students.
Current status: Completed.
Contact: Stephen Andrade, IRIS; (401) 863-2824.

Nutrition and Food Science

MacDiet

A nutrient analysis system that includes data on approximately 1,400 foods and calculates the RDA on specific nutrients and prints to final reports.
Current status: Classroom testing.
Contact: Coordinator, Microcomputer Program, Office of Computing Services, Drexel University, Philadelphia, PA 19104

Newdiet and Prodiet

These are Multiplan programs designed to solve nutritional composition problems involving linearly additive dietary elements as well as computation of the chemical score, an index of relative protein quality.
Current status: Available (Delphi and CompuServe).
Contact: Prof. Stephen Dubin, Department of Biomedical Engineering, Drexel University, Philadelphia, PA 19104

Philosophy

Venn, a Philosophy Tutor

This program allows the student to solve syllogisms by using Venn diagrams.
Current status: Available from Kinko's Academic Courseware Exchange (ACE).
Contact: ACE, 800-235-6919; or in California, 800-292-6640

Philosophy

Boole and Frege (for logic learning)

These two programs allow student practice and tutorial use of truth tables and natural deduction systems. While software for such purposes has existed for some time, this suite of programs aims to be easily usable by computer illiterates and to minimize the clerical work associated with proper proof procedures and presentation. Use of the keyboard is only required for input of assumptions and one's own problems. All other work is done by mouse manipulation. For example, the truth-table program draws the table and simply invites input of values. In natural deduction proofs, the student simply clicks on proof lines, clicks on the rule, clicks on DoIt, and gets a good line or a diagnosis of errors. In general, the system follows the conventions of Lemmon's beginning logic.

The programs are written in MacModula2 and embedded in them but modularised is a Macintosh interface that could be employed in other similar applications.
Current status: The truth-table program (Boole) and the propositional calculus natural deduction system (Frege) are complete. The predicate calculus system should be ready around October 1986. Send $7 for disk and postage.
Contact: David R. Bell, Department of Philosophy, The University GLASGOW, G12 8QQ, Scotland U.K.; (041) 339-8855

Physics

Interference Scanning Using ThunderScan

Current status: Unknown.
Contact: M. Andrews and J. Sandeman, Department of Physics, Australian National University

Interface with LeCroy Fast Digitizer Through IEEE 488 Bus

Current status: Unknown.
Contact: M. Andrews and J. Sandeman, Department of Physics, Australian National University

General Teaching Applications in Physics using Pascal Workshop

Current status: Unknown.
Contact: Dr. C. Edwards, Department of Physics, University of Western Australia

Physics

Elementary Particle Theory

Four programs (of a total of eight) have been developed by Prof. Savas Dimopoulos for a physics series on elementary particle theory and on symmetrics and Lagrangians.
Current status: Completed and continuing.
Contact: Prof. Savas Dimopoulos, Stanford University, Stanford, CA

Freshman Physics Lab

A set of data-acquisition tools, partially based on the Reed College DeskTop hardware, for the freshman physics lab.
Current status: Completed and in use.
Contact: Craig McGowan, Computer Center, University of Rochester, 727 Elmwood Avenue, Rochester, NY 14620

Physics Simulations

A series of simulations (based on rapid animation) to enable students to explore atomic theory, diffraction, electrostatics, harmonic motion, magnetism, monopoles, orbital physics, projectile motion, quantum mechanics, radiation, and special relativity without inappropriate or tedious calculations.
Current status: Available Kinko's Courseware Exchange; 800-235-6919, or in California, 800-292-6640
Contact: Professor Blas Cabrera, Department of Physics, Stanford University, Stanford, CA 94305

Animated Waves and Particles

Three animated graphics programs illustrate standing waves, traveling pulses, group velocity, sound waves, and electromagnetic waves; pressure, temperature, and gravity effects on an ideal gas, Brownian motion, diffusion through a hole, and electric and magnetic effects on moving charges; and the wave properties of an electron in an atom.
Current status: To be completed in fall 1986.
Contact: Eric T. Lane, Physics Department, The University of Tennessee at Chattanooga, Chattanooga, TN 37403-2598; (615) 755-4523

6502 Emulator

This program, written in BASIC, may be used for practice in assembly-language and machine-language programming of 6502 microprocessor-based systems. One window is used for writing and editing programs using assembly-language mnemonics. Another window shows the program listing. Other windows show and allow changes in the first 16 zero page memory locations, 16 other selected memory locations, and the processor status register. A step-mode window is used to step through a program with the effects on memory and the processor status register clearly displayed. A run mode is also available but is slow

6502 Emulator

because of the interpreter operation. Programs may be saved, and complete listings may be printed. Most of the 6502 instruction set is included. The program was written by David J. Zhu.
Current status: Completed.
Contact: Philip H. Sutter, Department of Physics, Franklin & Marshall College, Lancaster, PA 17604; (717) 291-4136

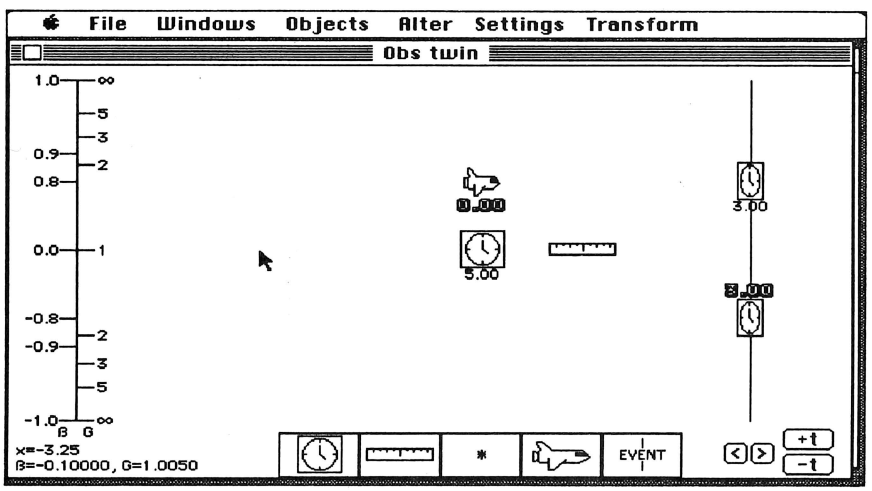
Chaos Primer

A collage of programs has been written that illustrates numerous classic examples of chaotic behavior. The Chaos Primer shows the evolution of attractors and strange attractors with fractal dimensionality. Extensive references are included.
Current status: Completed in summer 1986.
Contact: Dr. Russ Larsen, Department of Chemistry, Texas Tech University, Lubbock, TX 79409; (806) 742-3057

Modeling Toolkit in Special Relativity

Four "computer graphics toolkit" displays used by students to solve problems in special relativity. Embodying models of the world of the very fast, these interactive graphics utilities help students to visualize and develop an intuition for phenomena that cannot be experienced firsthand.

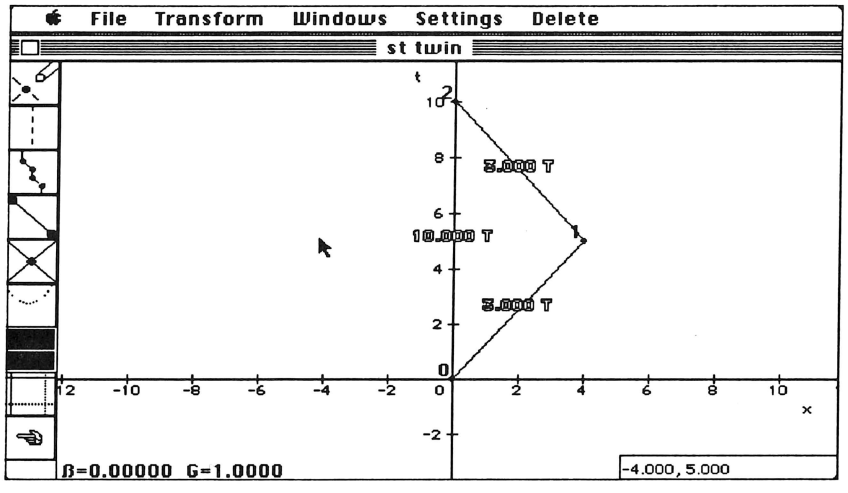
- OBSERVER. Clocks, rods, light flashes, and events are placed on a superhighway whose lanes represent different speeds. Motions and events are observed from the viewpoint of any clock or rod or from a shuttle that can change speeds and directions. Helps solve exercises exemplifying many difficult concepts in relativity, such as time stretching (Twin Paradox), Lorentz contraction, and relative synchronization of clocks.



Physics

Modeling Toolkit in Special Relativity

- SPACETIME DIAGRAM displays the locations of events in space and time on horizontal and vertical axes, respectively. The operator places light cones, invariant hyperbolas, straight worldlines, and the segmented world line of a shuttle that changes speeds and directions on the diagram. The program transforms the display to any reference frame in motion relative to the original frame.



OBSERVER and SPACETIME DIAGRAM share files to avoid double entry.

- MOMENTUM-ENERGY DIAGRAM analyzes collisions, creations, transformations, decays, and annihilations of particles moving in one spatial dimension. Displays momenta and energies of particles on horizontal and vertical axes, respectively. Separate BEFORE and AFTER displays allow matching of total energy and momentum before and after interaction to satisfy conservation laws. Displays can be transformed to other frames in relative motion, including the analytically important frame of zero total momentum.

- COLLISION provides a more formal analysis of the results of particle interactions in two spatial dimensions. The user enters on a table what is known about the masses, energies, and momenta of particles entering and leaving a collision. The computer calculates the unknown entries, displays results on three-dimensional perspective momentum-energy diagrams, and shows an animated "movie" of the interaction. Can also transform to the zero momentum frame of any particle or the frame of zero total momentum.

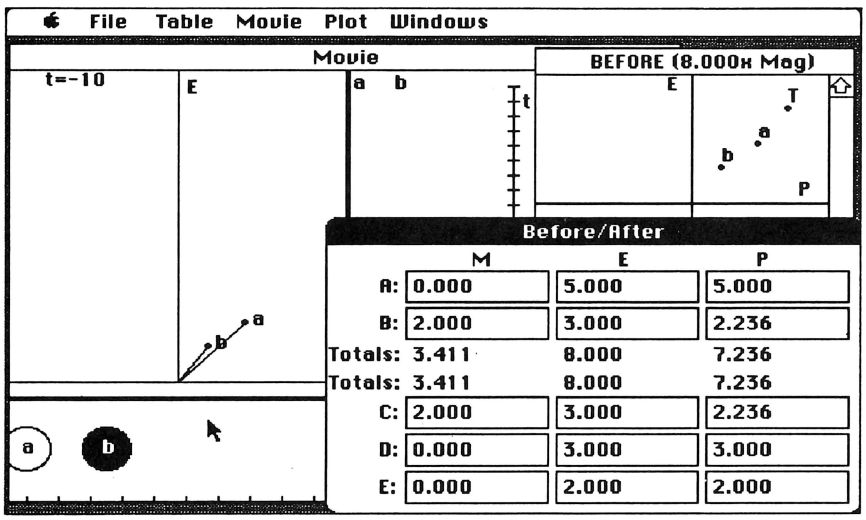
Physics

Modeling Toolkit in Special Relativity

COLLISION and MOMENTUM-ENERGY DIAGRAM share files for interactions in one spatial dimension.
Current status: Completed and continuing.
Contact: Edwin F. Taylor, Department of Physics, Massachusetts Institute of Technology, Cambridge, MA 02139

Elementary Quantum Mechanics

Five programs that compute the following: transmission/reflection coefficients and energy eigenvalues for bound states (in piecewise constant potentials); energy bands and density of states (in one-dimensional, piecewise, continuous periodic lattices); energy eigenvalues in arbitrary binding potentials.
Current status: Classroom testing.
Contact: Coordinator, Microcomputer Program, Office of Computing Services, Drexel University, Philadelphia, PA 19104



Psychology

Stimuli for Recognition Memory Experiments

In 1980, Snodgrass and Vanderwart published norms for 260 line drawings. Each drawing is rated on name agreement, image agreement, familiarity, and visual complexity. These stimuli have been digitized and stored in MacPaint format for easy use in memory experiments.
Current status: Completed. Price: $15 for two disks, payable to Rice University Psychology Department.
Contact: David M. Lane, Department of Psychology, Rice University, Houston, TX 77251-1892; (713) 527-8101 ext. 3412

Three Products

Machine-language library for use in psychology experiments, demonstration experiments in psychology, and geometrically rotated text.
Current status: Completed. Purchase price—all three disks for $15. Make checks payable to Rice University Psychology Department.
Contact: David M. Lane, Department of Psychology, Rice University, Houston, TX 77251-1892; (713) 527-8101 ext. 3412

Mac Laboratory Interfaces

Psychology Professor Allan Wagner is currently investigating use of the TAURUS system with MACQUISITION, a product of Small Business Computers of New England, in his animal learning lab.
Current status: Continuing. Version 1.2 available. Version 2 status unknown.
Contact: Prof. Allan Wagner, Yale University, c/o Box 220, Yale Station, New Haven, CT 06520

Experimentation Package for Human Information Processing Research

This program allows the user to carry out sophisticated experimental psychology research in the study of human information capabilities.
Current status: Unknown.
Contact: Salvador Algarabel, Universidad de Valencia, Facultad de Psicologia, Blasco Ibanel, 21, 46010, Valencia, Spain; (96) 3694850 ext. 213

Biopsychology

The Macintosh mouse and MacPaint software were used to track and record animal behaviors in an arena under different conditions of thermal and neuroendocrine manipulation.
Current status: Completed.
Contact: Timothy Schallert, University of Texas, Austin, TX 78712

Psychology

MacLaboratory for Psychology

Provides research-grade software and university-level interactional teaching materials. Total package consists of a laboratory manual designed for introductory or experimental psychology, five disks, and documentation. Most applications create an adjustable piece of psychological apparatus, providing tools for real research activities. There are six components:

- Part 1: Perceptual and Motor Skills. Various experimental settings permit the study of motor skill learning and eye-hand coordination using such tasks as mirror tracing and pursuit rotor tracking.

- Part 2: Problem Solving. Used to investigate the social interaction of group size with task difficulty.

- Part 3: Polygraph. Simulated GSR (Galvanic Silicone Response) that—with appropriate user interaction—can illustrate orienting response, habituation, classical conditioning, extinction, spontaneous recovery, partial reinforcement, and other concepts from the psychology of learning.

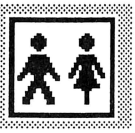
- Part 4: Pitch Discrimination. Waveform, frequency, amplitude, and duration can be modified and arranged into various sets for auditory testing.

- Part 5: Survey Questionnaires and Survey Data Summary. Applications facilitate the collection and simple summary of survey data.

- Part 6: Neuropsychology and Learning-Megaroids, an entertaining game, is used as an "icebreaker" and a source of data for instruction in experimental design. Also includes authoring software instructor can use to provide instruction in anatomy and physiology.

Current status: Completed. Available from Kinko's Courseware Exchange (ACE), 800-235-6919; in California, 800-292-6640.
Contact: Coordinator, Microcomputer Program, Office of Computing Services, Drexel University, Philadelphia, PA 19104

Sociology



SocStat: A Program to Computer Sociological Statistics

SocStat uses data sets from actual sociological surveys and allows the student to do elementary statistical analyses on them.
Current status: Completed. Send $4 for the disk.
Contact: The Personal Computing Center, McNutt Hall, Dartmouth College, Hanover, NH 03755

Telecommunications

Kermit

Developed by: Andrea Martin—Rice University
Kermit lets microcomputers and mainframe computers communicate and transfer files safely.
Current status: On hold.
Contact: Andrea Martin, ICSA, Rice University, Houston, TX 77251; (713) 527-4965

Macintosh Kermit

The Columbia University Center for Computing Activities completed a Kermit program that provides file transfer and terminal emulation for the Macintosh and Macintosh XL computers, allowing them to communicate with a large and diverse collection of mainframes, minicomputers, and microcomputers.
Current status: Available now.
Contact: Kermit Distribution, (212) 280-3703

SEAGate

The Stanford Ethernet—AppleTalk Gateway (SEAGate) is being implemented.
Current status: Complete.
Contact: Craig McGowan, Computer Center, University of Rochester, 727 Elmwood Avenue, Rochester, NY 14620

MacShell—A Sophisticated Terminal to UNIX

MacShell provides a sophisticated multiwindow interface to the UNIX operating system.
Current status: Unknown
Contact: Tim North, Department of Computer Science, University of Western Australia, Stirling Highway, Nedlands 6009, Western Australia

LaserTerminal: Serial-to-AppleTalk Gateway

LaserTerminal is a program developed at Reed College that makes the Macintosh function as a gate between an RS-232 serial line (for example, from a mainframe computer) and AppleTalk, enabling LaserWriter printing of PostScript documents.
Current status: Completed.
Contact: Marianne Colgrove, Education and Documentation Coordinator, Development Lab, Reed College, Portland, OR 97202

Griffin Terminal 100

Griffin Terminal 100 (GT100), developed at Reed College, is a terminal-emulation application for the Macintosh that can function in either Tektronix 4012 or VT100 mode, and can handle technical graphics. GT100 is available

through Metaresearch, Inc., 1100 SE Woodward, Portland, OR 97202; (503) 232-1712. Price: $99 (academic discounts available).
Current status: Completed.
Contact: Scott Gillespie, Software Development Manager, Development Lab, Reed College, Portland, OR 97202; (503) 771-1112 ext. 277

Griffin Terminal 100

Current status: Complete.
Contact: J. Radnell and P. Wesely-Smith, Department of Asian Studies, Australian National University

Communications Link with Sigma Data WP units

Current status: Unknown.
Contact: J. Radnell and P. Wesely-Smith, Department of Asian Studies, Australian National University

Communications Link with Epson WP units

Current status: Unknown.
Contact: K. Loane, Information Science, University of Tasmania

Macintosh Videotext Terminal Emulation

Current status: Unknown.
Contact: David Hawkins, Computer Science, Australian National University

MacWriter-to-Scribe Formatter

Tincan is a terminal-emulation and file transfer program for the Apple Macintosh. With Tincan, your Macintosh can be connected to a host computer (mainframe), and used as if it were a terminal normally used to connect to a mainframe timesharing service. Instead of logging on through a DEC VT100 or Datamedia 1520 terminal, you can use the Macintosh to log on to the IBM's VM/CMS or MVS/TSO, DEC's VAX/VMS, or a UNIX 4.2. In addition, Tincan is designed to operate well in conjunction with the Yale ASCII Communications System (an IBM Series/1 with Yale ASCII software, the IBM 4994, or the IBM 7171), a product that allows ASCII terminals to access IBM mainframes as IBM 3270-style full-screen terminals.

If you have any implementation of Yale ASCII attached to your mainframe, then in conjunction with Tincan on the Macintosh, you get IBM 3270 support, with some enhancements. By installing PCTRANS (a host command available for a nominal charge from the Yale Computer Center), Tincan can perform error-free file transfer for the movement of Macintosh text files, binary data files, or entire Macintosh applications (application transfer utilizes the MacBinary protocol) between Macintosh and IBM VM/CMS or MVS/TSO, DEC VAX/VMS, or UNIX 4.2 services.

Tincan: A Terminal Emulator for the Macintosh

Telecommunications

Tincan: A Terminal Emulator for the Macintosh

The current version (1.01) has been in distribution for a while. Version 2.0 was scheduled for a fall release. Some of Version 2.0's main features include:

- Runs on Macintosh 512K, Macintosh Plus under HFS, and Macintosh XL (not recommended for the Macintosh 128K).
- Supports up to four services (hosts), selectable via dialogs or configuration files (Tincan documents).
- Provides error-free file transfer to and from the remote computer via PCTRANS.
- Offers automatic and semiautomatic log ons using a special-purpose scripting facility.
- Implements almost all features, including key assignments, screen buttons, and communications parameters, as resources that can be modified with custom Resedit templates supplied with Tincan.
- Displays on-line help for all features, including buttons and dialogs, with a simple Option-click on the item.
- Includes a map showing all key assignments.
- Offers direct cursor positioning using the mouse.
- Features on-screen buttons supporting common functions.
- Implements 36 PF keys as Escape sequences (the ~ key is mapped to ESC); the 12 keys in each of the top three rows invoke PF1 through PF36 when preceded by ESC.
- Uses Option as a second shift for the PA, cursor, and extended word-processing functions.
- Provides a session capture with a scrolling window for TTY communications.
- Allows simple file send and receive for TTY sessions.
- Offers quick print for plain ASCII files.
- Supports printing on ImageWriter and LaserWriter printers.

Tincan was developed and written at Yale specifically for Yale students, faculty, and staff. It is sold to non-Yale affiliated individuals for $50 prepaid, including the program(non copy-protected) a user's guide and a set of Tincan documents—sample configuration files set up for the commonly used services at Yale. Site licenses are available for $1,000 to educational institutions and for $2,000 to commercial sites. YTERM, a similar program for the IBM PC, also developed at Yale, is available under the same conditions. Note that Tincan site licenses are separate from YTERM site licenses and the purchase of one does **not** entitle the customer to the other. The site license grants copying rights to the manual and disk, and one year's updates, for use by members of the licensing institution. You may purchase it by writing.

Current status: Completed.

Contact: Yale Computer Center, 175 Whitney Avenue, P.O. Box 2112., Yale Station, New Haven, CT 06520-2112

Word Processing

MacWrite Translator

The translator consists of two programs that allow you to translate a document from MacWrite Version 3 (RAM-based) format to that of another word processor.
Current status: Public domain software available now.
Contact: DCS, 401 URIS Hall, Cornell University, Ithaca, NY 14853

Griffin Text

GriffinText is a fast, no-frills text processor developed at the Reed College MacLab.
Current status: Available. Consortium price: $39. Retail price: $69.
Contact: Prof. Richard Crandall, Reed College, Portland, OR 97202; (503) 771-1112 ext. 277

LaserWriter Driver for IBM PC Scientific Word Processor

Current status: Unknown
Contact: Neville Smythe, Department of Mathematics, Australian National University

Indexer for MacWrite

This project allows automatic or interactive indexing/cross-referencing/keyboarding.
Current status: Completed.
Contact: Neville Smythe, Department of Mathematics, Australian National University

Library Word-Processing Center

The Harold B. Lee Library at Brigham Young University has a word-processing center that includes (among other things) 10 Macintosh computers and 2 LaserWriter printers. The Macintosh computers are available for $1.50/hr. and the LaserWriter printers for $3.00/hr. to students and faculty and include word-processing software. They are quite popular: each LaserWriter printer averages 15,000 pages per month!
Current status: Completed.
Contact: Scott Eldredge, Manager, HBLL Word Processing Center, 3361 HBLL, Brigham Young University, Provo, UT 84604; (801) 378-6723

Apple Computer, Inc.
20525 Mariani Avenue
Cupertino, California 95014
408 996-1010
TLX 171-576

M1517

Apple Macintosh WHEELS FOR THE MIND WINTER 1987 Vol.3 No.1